

# Faster Approximate Multicommodity Flow Using Quadratically Coupled Flows \*

Jonathan A. Kelner <sup>†</sup>  
MIT  
kelner@mit.edu

Gary L. Miller  
CMU  
glmiller@cs.cmu.edu

Richard Peng <sup>‡</sup>  
CMU <sup>§</sup>  
yangp@cs.cmu.edu

May 9, 2012

## Abstract

The maximum multicommodity flow problem is a natural generalization of the maximum flow problem to route multiple distinct flows. Obtaining a  $1 - \epsilon$  approximation to the multicommodity flow problem on graphs is a well-studied problem. In this paper we present an adaptation of recent advances in single-commodity flow algorithms to this problem. As the underlying linear systems in the electrical problems of multicommodity flow problems are no longer Laplacians, our approach is tailored to generate specialized systems which can be preconditioned and solved efficiently using Laplacians. Given an undirected graph with  $m$  edges and  $k$  commodities, we give algorithms that find  $1 - \epsilon$  approximate solutions to the maximum concurrent flow problem and the maximum weighted multicommodity flow problem in time  $\tilde{O}(m^{4/3} \text{poly}(k, \epsilon^{-1}))$  <sup>1</sup>.

---

\*Partially supported by the National Science Foundation under grant number CCF-1018463.

<sup>†</sup>Partially supported by NFS Awards 0843915 and 1111109

<sup>‡</sup>Supported by a Microsoft Fellowship

<sup>§</sup>Part of this work was done while at Microsoft Research New England

<sup>1</sup>We use  $\tilde{O}(f(m))$  to denote  $\tilde{O}(f(m) \log^c f(m))$  for some constant  $c$

# 1 Introduction

The multicommodity flow problem is a natural extension of the maximum flow problem. One of its variations, maximum concurrent multicommodity flow, asks to route multiple demands simultaneously in a network subject to capacity constraints. In this setting we're given an undirected, capacitated graph  $G = (V, E, \mathbf{u})$  where  $\mathbf{u} : E \rightarrow \mathbb{R}^+$ , and  $k$  source-sink pairs  $(s_1, t_1) \dots (s_k, t_k)$ . The goal is to find the maximum  $\lambda$  such that there exist  $k$  flows  $\mathbf{f}_1 \dots \mathbf{f}_k$  where  $\mathbf{f}_i$  routes  $\lambda$  units of flow between  $s_i$  and  $t_i$ , and the total flow along each edge obey the following capacity constraint:

$$\sum_i |\mathbf{f}_i(e)| \leq \mathbf{u}(e) \quad \forall e \in E$$

## 1.1 Related Work

The simplest version of the problem is with two commodities in an undirected graph. In this case the problem was shown to be reducible to two single commodity maximum flow problems [RW66]. When there are 3 or more commodities though, this connection no longer holds and all of the (almost) exact algorithms for multicommodity flow problems involve solving a linear programming formulation. For these linear programs, the method with the best asymptotic behavior is the interior point algorithm, which requires solving  $O(m^{1/2})$  linear systems. By tracking inverses of these systems and making low rank updates, Vaidya showed an algorithm with running time  $\tilde{O}(k^{5/2}m^{3/2}n)$  [Vai89]. This is not very far from the natural barrier of  $\tilde{\Omega}(k^{1/2}m^{1/2}n^2)$  for this type of approach, which arises from the need to compute a dense matrix-vector product involving the inverse in each iteration.

Subsequent work on multicommodity flow focused on obtaining  $1 + \epsilon$  approximate solutions in faster time. This work initially focused on the case of small  $k$ , and the algorithms are based on solving multiple minimum cost flow problems [LMP<sup>+</sup>91]. When combined with the minimum cost flow algorithm from [DS08], these algorithms gave a running time of  $\tilde{O}(m^{1.5}\text{poly}(k, \epsilon^{-1}))$ .

More recent approaches have favored using a less expensive inner loop to obtain better bounds when  $k$  is large. Specifically multiplicative weights update method using single source shortest path routines as oracles [GK98, Fle00]. The most recent among these approaches obtained a running time of  $\tilde{O}(nm/\epsilon^2)$  using dynamic graph data structures [Mad10]. These methods give better performance the case where  $k$  is large. However, when applied to instances with a smaller value of  $k$  these approaches encounter similar issues to those encountered by path based single-commodity flow algorithms: the flow decomposition barrier at  $\Omega(nm)$ . This barrier stems from the fact that if we decompose a flow into a list of paths of paths, the total size of these paths can be  $\Omega(nm)$ .

An alternate approach to solving linear systems, has led to a possible way to circumvent both the dense inverse and flow decomposition barrier. The graph-like nature of the underlying linear system, graph Laplacians, allows one to find sparse approximations of it, which, when used to precondition iterative solvers, led to speedups in solving such systems [Vai91]. This approach using graph Laplacians has been extended greatly in subsequent works, leading to algorithms with nearly-linear running times [ST06]. Graph Laplacians are closely connected with problems involving a single flow, such as maximum flow, minimum cost flows and shortest path. To the best of our knowledge this connection was first observed in [DS08], leading to, among others, a faster algorithm for minimum cost flow. This algorithm is used as a subroutine in the  $\tilde{O}(m^{1.5}\text{poly}(k, \epsilon^{-1}))$  algorithm mentioned earlier.

## 1.2 Our Work

Recently, the running time for approximate maximum flow in undirected graphs has been improved to  $\tilde{O}(m^{4/3})$  [CKM<sup>+</sup>11]. A natural question arising from this is whether this algorithm can also be extended to multicommodity flow. Even though in the 2-commodity setting their algorithm can be invoked in a

black-box manner [RW66], a more general examination of this setting is helpful for understanding the main components of our extensions.

In order to further simplify the 2-commodity case, we assume that each edge have unit capacity. That is, we want to find two flows  $\mathbf{f}_1$  and  $\mathbf{f}_2$  that meet their respective demands, and satisfy the following capacity constraint on each edge  $e$ .

$$|\mathbf{f}_1(e)| + |\mathbf{f}_2(e)| \leq 1$$

One way to visualize this constraint is by considering each flow assignment as a coordinate in the 2-D plane. Then a point  $(\mathbf{f}_1(e), \mathbf{f}_2(e))$  obeys this constraint if it's inside the unit  $L_1$  ball, as shown in Figure 1. The Christiano et al. algorithm [CKM<sup>+</sup>11] produces a flow that approximately satisfies edge capacities by solving a series of electrical problems. To form such electrical problems, they assign one resistor per edge, leading to a quadratic term of the form  $\mathbf{f}(e)^2$ . Since  $\mathbf{f}(e) \leq 1$  is equivalent to  $\mathbf{f}(e)^2 \leq 1$ , they're able to bound the energy of each of these terms by 1. A natural generalization to two commodities would be to bound the sum of the squares of the two flows, leading to an instance of what we define as **Quadratically Capacitated Flows**. Specifically we would like the following to hold along each edge:

$$\mathbf{f}_1(e)^2 + \mathbf{f}_2(e)^2 \leq 1$$

Note that the flow  $(\mathbf{f}_1(e), \mathbf{f}_2(e)) = (1, 0)$  has energy 1, so the RHS value of 1 is tight. This corresponds to allowing any  $(\mathbf{f}_1(e), \mathbf{f}_2(e))$  that's within the unit  $L_2$  ball, and as shown in Figure 1. However, in the 2-commodity case it's possible for flow settings that over-congest the edge to still satisfy this energy constraint. In other words, it's possible for a point to be in the unit  $L_2$  ball but outside the unit  $L_1$  ball. For example, the flow  $(\mathbf{f}_1(e) = \sqrt{2}/2, \mathbf{f}_2(e) = \sqrt{2}/2)$  also meets this energy constraint despite having a congestion  $\sqrt{2}$ .

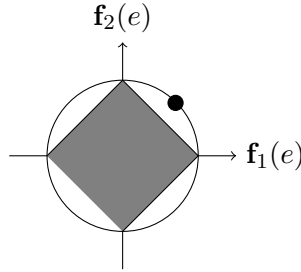


Figure 1: Unit  $l_1$  ball representing region of feasible flows (gray), with unit  $l_2$  ball being the quadratic capacity constraint. The point  $(\mathbf{f}_1(e) = \sqrt{2}/2, \mathbf{f}_2(e) = \sqrt{2}/2)$  obeys this constraint, but exceeds the edge's capacity

One possible remedy to this problem is to introduce more intricate quadratic coupling between the two commodities. However, we still need to set the constraint so that any flow whose total congestion is below the capacity falls within this ellipse. This is equivalent to the ellipse containing the unit  $L_1$  ball, which along with the fact that ellipses have smooth boundaries means we must allow some extra points. For example, the modified ellipse in Figure 2 once again allows the returned solution to have large congestion. In general, computing a single quadratically capacitated flow can lead to the edge being over-congested by a factor of  $\sqrt{k}$ , giving a  $\sqrt{k}$  approximation.

As a result, instead of computing a single quadratically capacitated flow, we compute a sequence of them and **average** the result. Note that although both  $(\mathbf{f}_1(e) = \sqrt{2}/2, \mathbf{f}_2(e) = \sqrt{2}/2)$  and  $(\mathbf{f}_1(e) = \sqrt{2}/2, \mathbf{f}_2(e) = -\sqrt{2}/2)$  have congestion of  $\sqrt{2}$ , if we average them we're left with a flow with congestion only  $\sqrt{2}/2$ . If we're only concerned with keeping the average congestion small, the flows computed in previous iterations can give us some 'slack' in certain directions. As it turns out, if we compute the coupling matrix based on the flows returned so far, it's possible to move the average gradually get closer to the unit  $L_1$  ball. For example, the average of the two flows returned in Figure 2 is inside the unit  $L_1$  ball despite both falling outside of it.

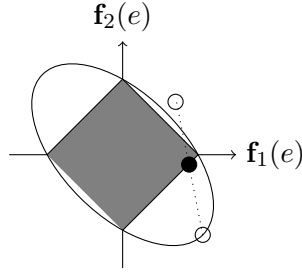


Figure 2: Modified energy constraint after a flow of  $(\sqrt{2}/2, \sqrt{2}/2)$  has been added. Note that the point returned is still outside of the unit  $l_1$  ball, but the average is within it.

The problem now becomes finding feasible quadratically capacitated flows. The Christiano et al. algorithm [CKM<sup>+</sup>11] can be adapted naturally to this problem, providing that we can find flows that minimizes a weighted sum of the energy terms. We define this generalization of electrical flows as **quadratically coupled flows**. Just like their single commodity version, the minimum energy quadratically coupled flows can also be computed by solving linear systems.

The remaining difficulty of the problem is now with solving these linear systems. Most of the combinatorial preconditioning framework relies on the system being decomposable into 2-by-2 blocks corresponding to single edges. For quadratically coupled flows, the resulting systems are only decomposable into  $2k$ -by- $2k$  blocks. These systems are also encountered in stiffness matrices of finite element systems [BHV04], and  $k$ -dimensional trusses [DS07, AT11]. To date a nearly-linear time solver that can handle all such systems remains elusive.

Instead of solving these systems directly, we show, by more careful analysis of our algorithm that generates the quadratically capacitated flow problems, that it suffices to consider a more friendly subset of them. It can be shown that the  $\sqrt{k}$  factor deviation that occurs in the uncoupled electrical problem is also the extent of our loss if we try to approximate these more friendly quadratically coupled flows with uncoupled ones. However, in the quadratic case such losses are fixable using preconditioned iterative methods, allowing us to solve the systems arising from quadratically coupled flows by solving a number of graph Laplacians instead. This leads us our main result, which can be stated as:

**Theorem 1.1** *Given an undirected, capacitated graph  $G = (V, E, \mathbf{u})$  with  $m$  edges, along with  $k$  commodities and their demands  $\mathbf{d}_1 \dots \mathbf{d}_k$ . There is an algorithm that computes an  $1 - \epsilon$  approximate maximum concurrent flow in time:*

$$\tilde{O}(m^{4/3} \text{poly}(k, \epsilon^{-1}))$$

An overview of the main steps of the algorithm is shown in Section 3. Our approach also extends to maximum weighted multicommodity flow, which we show in Appendix B.

## 2 Preliminaries

The maximum concurrent multicommodity problem concerns the simultaneous routing of various commodities in a capacitated network. For our purposes, the graph is an undirected, capacitated graph  $G = (V, E, \mathbf{u})$  where  $\mathbf{u} : E \rightarrow \mathbb{R}^+$  is the capacity of each edge. If we assign an arbitrary orientation to the edges, we can denote the edge-vertex incidence matrix  $\mathbf{B} \in \mathbb{R}^{m \times n}$  as:

$$\mathbf{B}(e, u) = \begin{cases} 1 & \text{if } u \text{ is the head of } e \\ -1 & \text{if } u \text{ is the tail of } e \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

Then, for a (single commodity) flow  $\mathbf{f}$ , the excess of the flow at each vertex is given by the length  $n$  vector  $\mathbf{B}^T \mathbf{f}$ .

Throughout the paper we let  $k$  be the number of commodities routed. It can be shown that it suffices to solve the  $k$ -commodity flow problem for fixed vertex demands  $\mathbf{d}_1, \mathbf{d}_2 \dots \mathbf{d}_k$  one for each commodity. The goal of finding a flow that concurrently routes these demands in turn becomes finding  $\mathbf{f}_i$  for each commodity such that:

$$\mathbf{B}^T \mathbf{f}_i = \mathbf{d}_i \quad (2.2)$$

The other requirement for a valid flow is that the flows cannot exceed the capacity of an edge. Specifically we need the following constraint for each edge  $e$ :

$$\sum_{1 \leq i \leq k} |\mathbf{f}_i(e)| \leq \mathbf{u}(e) \quad (2.3)$$

### 2.1 Notations for $k$ -Commodity Flow and Vertex Potentials

The extension of a single variable indicating flow/vertex potential on an edge/vertex to  $k$  variables creates several notational issues. We use a length  $km$  vector  $\mathbf{f} \in \mathbb{R}^{km}$  to denote a  $k$ -commodity flow, and allow for two ways to index into it based on commodity/edge respectively. Specifically, for a commodity  $i$ , we use  $\mathbf{f}_i$  to denote the length  $m$  vector with the flows of commodity  $i$  along all edges and for an edge  $e$ , we use  $\mathbf{f}(e)$  to denote the length  $k$  vector containing the flows of all  $k$  commodities along this edge.

This definition extends naturally to vectors over all (vertex, commodity) pairs as well. We let  $\mathbf{d} \in \mathbb{R}^{nk}$  be the column vector obtained by concatenating the length  $k$  demand vector over all  $n$  vertices. If the edges are labeled  $e_1 \dots e_m$  and the vertices  $v_1 \dots v_n$ , then  $\mathbf{f}$  and  $\mathbf{d}$  can be written as:

$$\mathbf{f}^T = [\mathbf{f}(e_1)^T, \mathbf{f}(e_2)^T, \dots, \mathbf{f}(e_m)^T] \quad (2.4)$$

$$\mathbf{d}^T = [\mathbf{d}(v_1)^T, \mathbf{d}(v_2)^T, \dots, \mathbf{d}(v_n)^T] \quad (2.5)$$

We can also define larger matrices that allows us to express these conditions across all  $k$  commodities, and their interactions more clearly. The edge-vertex incidence matrix that maps between  $\mathbf{d}$  and  $\mathbf{f}$  is the Kronecker product between  $\mathbf{B}$  and the  $k \times k$  identity matrix  $\mathbf{I}_k$ .

$$\Gamma = \mathbf{B} \otimes \mathbf{I}_k \quad (2.6)$$

and  $\mathbf{f}$  meeting the demands can be written as:

$$\Gamma^T \mathbf{f} = \mathbf{d} \quad (2.7)$$

Note that flows of two commodities passing in opposite directions through the edge do not cancel each other out.

By obtaining crude bounds on the flow value using bottle neck shortest paths and binary searching in the same way as in [CKM<sup>+</sup>11], the maximum concurrent multicommodity flow problem can be reduced to  $O(\log n)$  iterations of checking whether there is a  $k$ -commodity flow  $\mathbf{f}$  that satisfies the following:

$$\begin{aligned} \|\mathbf{f}(e)\|_1 &= \sum_{i=1}^k |\mathbf{f}_i(e)| \leq u(e) & \forall e \in E \\ \Gamma \mathbf{f} &= \mathbf{d} \end{aligned}$$

## 2.2 Quadratic Generalizations

We define two generalizations of electrical flows to multiple commodities. Our main goal is to capture situations where the amount of flows of one type allowed on an edge depends inversely on the amount of another flow, so the flows are “coupled.” To do so, we introduce a positive-definite, block-diagonal matrix  $\mathbf{P} \in \Re^{km \times km}$  such that  $\mathbf{P} = \sum_e \mathbf{P}(e)$  and each  $\mathbf{P}(e)$  is a  $k \times k$  positive definite matrix defined over the  $k$  entries corresponding to the flow values on edge  $e$ .

For each edge the  $k$  flows along an edge  $e$ ,  $\mathbf{f}(e)$  we get a natural quadratic penalty or energy dissipation term:

$$\mathcal{E}_{\mathbf{f}}(\mathbf{P}, e) = \mathbf{f}(e)^T \mathbf{P}(e) \mathbf{f}(e) \quad (2.8)$$

Summing these gives the total energy dissipation of a set of flows, denoted using  $\mathcal{E}$ .

$$\begin{aligned} \mathcal{E}_{\mathbf{f}}(\mathbf{P}) &= \sum_e \mathcal{E}_{\mathbf{f}}(\mathbf{P}, e) \\ &= \mathbf{f}^T \mathbf{P} \mathbf{f} \end{aligned} \quad (2.9)$$

In the **Quadratically Coupled Flow** problem, we aim to find a flow  $\mathbf{f}$  that satisfies all of the demand constraints and minimizes the total energy dissipation, namely:

$$\min \quad \mathcal{E}_{\mathbf{f}}(\mathbf{P}) \quad (2.10)$$

$$\text{subject to:} \quad \Gamma^T \mathbf{f} = \mathbf{d} \quad (2.11)$$

The minimum is denoted by  $\mathcal{E}(\mathbf{P})$ . We can define the related potential assignment problem, where the goal is to assign potentials to the vertices to separate the demands. Note that due to there being  $k$  commodities, a potential can be assigned to each (flow, vertex) pair, creating  $\phi \in \Re^{kn}$ . This vector can also be viewed as being composed of  $n$  length  $k$  vectors, with the vector at vertex  $u$  being  $\phi(u)$ . Given an edge  $e = (u, v)$  whose end points connects vertices with potentials  $\phi(u)$  and  $\phi(v)$ , the difference between its end points is  $\phi(u) - \phi(v)$ . In order to map this length  $nk$  vector into the same support as the  $k$ -commodity

flows along edges, we need to multiply it by  $\Gamma$ , and we denote the resulting vector as  $\mathbf{y}$ :

$$\mathbf{y} = \Gamma\phi \quad (2.12)$$

The energy dissipation of an edge with respect to  $\phi$  can in turn be defined as:

$$\mathcal{E}_\phi(\mathbf{P}, e) = \mathbf{y}\mathbf{P}(e)^{-1}\mathbf{y} \quad (2.13)$$

Note that this definition relies on  $\mathbf{P}(e)$  being positive definite and therefore invertible on the support corresponding to edge  $e$ . This can in turn be extended analogously to the energy dissipation of a set of potentials as:

$$\begin{aligned} \mathcal{E}_\phi(\mathbf{P}) &= \sum_e \mathcal{E}_\phi(\mathbf{P}, e) \\ &= \phi^T \Gamma^T \mathbf{P}^{-1} \Gamma \phi \end{aligned} \quad (2.14)$$

We further generalize the definition of a Laplacian to  $k$  commodities:

$$\mathcal{L} = \Gamma^T \mathbf{P}^{-1} \Gamma \quad (2.15)$$

Thus, the energy dissipation of a set of potentials also equals to  $\phi^T \mathcal{L} \phi$ . Which leads to the following maximization problem, which is the dual of the quadratically coupled flow problem.

$$\max \quad (\mathbf{d}^T \phi)^2 \quad (2.16)$$

$$\text{subject to:} \quad \mathcal{E}_\phi(\mathbf{P}) \leq 1 \quad (2.17)$$

We denote the optimum of this value using  $\mathcal{C}_{eff}(\mathbf{P})$  and will show in Section 4.1 that  $\mathcal{C}_{eff}(\mathbf{P}) = \mathcal{E}(\mathbf{P})$ .

Another coupled flow problem that's closer to the maximum concurrent flow problem is one where we also bound the **saturation** of them w.r.t.  $\mathbf{P}$ , where **saturation** is the square root of the energy dissipation.

$$\text{saturation}_f(\mathbf{P}, e) = \sqrt{\mathbf{f}(e)^T \mathbf{P}(e) \mathbf{f}(e)} \quad (2.18)$$

Finding a flow with bounded saturation per edge will be then called the **Quadratically Capacitated Flow** problem.

### 3 Overview of Our Approach

A commonality of the algorithms for flow with Laplacian solves as an inner loop [DS08, CKM<sup>+</sup>11] is that they make repeated computations of an optimum electrical flow in a graph with adjusted edge weights. The main problem with extending these methods to  $k$ -commodity flow is that the Laplacian for  $k$ -commodity electrical flow  $\mathcal{L}$  is no longer symmetrically diagonally dominant. Our key observation in resolving this issue is that when the energy matrices  $\mathbf{P}(e)$  are well-conditioned, we can precondition  $\mathbf{P}$  with a diagonal matrix. Then using techniques similar to those in [BHV04], we can solve systems involving  $\mathcal{L}$  using a small number of Laplacian linear system solves. Our algorithm for  $k$ -commodity flow has the following layers with a description in Figure 3 as well.

1. We adapt the algorithm from [LMP<sup>+</sup>91] to use flows with electrical capacity constraints associated with the  $k$  commodities instead of minimum cost flow as its oracle call. At the outermost level, the approximately multi-commodity flow algorithm, repeatedly computes a positive definite matrix  $\mathbf{P}(e)$  for each edge based on the flows on it so far on that edge, and boost their diagonal entries to keep their condition number at most  $\mathbf{poly}(k)$ . The outermost level then calls an algorithm that computes quadratically capacitated flow that is:

$$\mathbf{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}, e) \leq \max_{\mathbf{f}, \|\mathbf{f}(e)\|_1 \leq \mathbf{u}(e)} \mathbf{saturation}_{\mathbf{f}}(\mathbf{P}, e) \quad (3.19)$$

After repeating this process  $\mathbf{poly}(k)$  times, averaging these flows gives one where  $\|\tilde{\mathbf{f}}(e)\|_1 \leq (1+\epsilon)\mathbf{u}(e)$  on all edges. We give two methods for computing  $\mathbf{P}$  in Sections 6 and 7.

2. We use an algorithm that's a direct extension of the electrical flow based maximum flow algorithm from [CKM<sup>+</sup>11] to minimize the maximum saturation of an edge. This stage of the algorithm in turn solves  $\tilde{O}(m^{1/3})$  quadratically coupled flows where the energy coupling on an edge is  $\tilde{\mathbf{P}}(e) = \mathbf{w}_e \mathbf{P}(e)$ . Note that since  $\mathbf{P}(e)$  was chosen to be well conditioned and  $\mathbf{w}_e$  is a scalar, the  $\tilde{\mathbf{P}}(e)$ s that we pass onto the next layer on remains well-conditioned. This is presented in Section 5.
3. In turn the Quadratically Capacitated Flow Algorithm makes calls to an algorithm that computes a quadratically coupled flow. The almost-optimal quadratically coupled flow is obtained by linear solves involving  $\mathcal{L}$ . Specifically, we show that preconditioning each  $P(e)$  with a diagonal matrix allows us to decouple the  $k$ -flows, at the cost of a mild condition number set in the outermost layer. Then using preconditioned Chebyshev iteration, we obtain an almost optimal quadratically coupled flow using  $\mathbf{poly}(k, \epsilon^{-1})$  Laplacian solves on a matrix with  $m$  non-zero entries. Properties of the  $k$ -commodity electrical flow, as well as bounds on the error and convergence of the solves are shown in Section 4.

## 4 Approximate Computation of Quadratically Coupled Flows

### 4.1 Quadratically Coupled Flows and Vertex Potentials

Let  $\bar{\mathbf{x}}$  be the vector such that  $\mathcal{L}\bar{\mathbf{x}} = \mathbf{d}$ . It can be shown that  $\mathcal{E}(\mathbf{P}, \phi)$  is maximized when  $\phi$  is a multiple of  $\bar{\mathbf{x}}$ . Then the scaling quantity  $\bar{\lambda}$  as well as the optimum set of potential  $\bar{\phi}$  are:

$$\bar{\lambda} = \sqrt{\mathbf{d}^T \mathcal{L}^+ \mathbf{d}} \quad (4.20)$$

$$\begin{aligned} \bar{\phi} &= \frac{1}{\bar{\lambda}} \bar{\mathbf{x}} \\ &= \frac{1}{\bar{\lambda}} \mathcal{L}^+ \mathbf{d} \end{aligned} \quad (4.21)$$

Note that  $\mathbf{d}$  satisfies  $\mathbf{1}_i^T \mathbf{d} = 0$  for all  $1 \leq i \leq k$ . Also, since  $\mathbf{P}$  is positive-semidefinite, the null space of  $\mathcal{L}$  is precisely the space spanned by the  $k$  vectors  $\mathbf{1}_i$ . Therefore  $\mathbf{d}$  lies completely within the column space of  $\mathcal{L}$  and we have  $\mathcal{L}\mathcal{L}^+ \mathbf{d} = \mathbf{d}$ . The value of  $\mathbf{d}^T \bar{\phi}$  is then:

$$\begin{aligned} \mathbf{d}^T \bar{\phi} &= \mathbf{d}^T \frac{1}{\bar{\lambda}} \mathcal{L}^+ \mathbf{d} \\ &= \bar{\lambda} \end{aligned} \quad (4.22)$$



---

MAXCONCURRENTFLOW

**Constraint on desired flow:** For each  $e$ , total flow  $|f_e| \leq 1$ .

Repeatedly updates energy matrices using matrix multiplicative weights. Makes  $\text{poly}(k)$  oracle calls to:

QUADRATICALLYCAPACITATEDFLOW

**Constraint on desired flow:** For each  $e$ ,  $\text{saturation}_{\mathbf{f}}(\mathbf{P}, e) = \sqrt{\mathbf{f}(e)^T \mathbf{P}(e) \mathbf{f}(e)} \leq 1$ .

Repeatedly updates energy matrices using (scalar) multiplicative weights. Makes  $\tilde{O}(m^{1/3})$  oracle calls to:

QUADRATICALLYCOUPLEDFLOW

**Constraint on desired flow:** Minimize total dissipated energy  $\mathcal{E}_{\mathbf{f}}(\mathbf{P}) = \sum_e \mathbf{f}(e)^T \mathbf{P}(e) \mathbf{f}(e)$ .

Solves 1 linear system using:

PRECONCHEBY

Solves non-Laplacian system by preconditioning with  $k$   $n \times n$  Laplacians. Solves these using  $k$  calls to nearly-linear time Laplacian solvers.

Figure 3: The high-level structure of the algorithm and the approximate number of calls made to each routine (for fixed  $\epsilon$ ).

The optimal quadratically coupled flow can be obtained from the optimal vertex potentials as follows:

$$\begin{aligned} \bar{\mathbf{f}} &= \mathbf{P}^{-1} \Gamma \bar{\mathbf{x}} \\ &= \bar{\lambda} \mathbf{P}^{-1} \Gamma \bar{\phi} \end{aligned} \tag{4.23}$$

We can prove the following generalizations of standard facts about electrical flow/effective resistance for multicommodity electrical flows.

**Fact 4.1**    1.  $\bar{\mathbf{f}}$  satisfies the demands, that is  $\Gamma^T \bar{\mathbf{f}} = \mathbf{d}$ .

2.  $\mathcal{E}(\mathbf{P}, \bar{\mathbf{f}}) = \mathbf{d}^T \mathcal{L}^+ \mathbf{d}$

3. For any other flow  $\mathbf{f}$  that satisfies the demands,  $\mathcal{E}(\mathbf{P}, \mathbf{f}) \geq \mathcal{E}(\mathbf{P}, \bar{\mathbf{f}})$ .

**Proof**

Part 1

$$\begin{aligned} \Gamma^T \bar{\mathbf{f}} &= \bar{\lambda} \Gamma^T \mathbf{P} \Gamma \bar{\phi} \\ &= \bar{\lambda} \mathcal{L} \frac{1}{\bar{\lambda}} \mathcal{L}^+ \mathbf{d} \\ &= \mathbf{d} \end{aligned} \tag{4.24}$$

Part 2

$$\begin{aligned}
\mathcal{E}(\mathbf{P}, \bar{\mathbf{f}}) &= \bar{\mathbf{f}}^T \mathbf{P} \bar{\mathbf{f}} \\
&= \mathbf{d}^T \mathcal{L}^+ \mathbf{d} (\mathbf{P}^{-1} \Gamma \bar{\phi})^T \mathbf{P} (\mathbf{P}^{-1} \Gamma \bar{\phi}) \\
&= \mathbf{d}^T \mathcal{L}^+ \mathbf{d} \bar{\phi}^T \Gamma^T \mathbf{P}^{-1} \mathbf{P} \mathbf{P}^{-1} \Gamma \bar{\phi} \\
&= \mathbf{d}^T \mathcal{L}^+ \mathbf{d} \bar{\phi}^T \mathcal{L} \bar{\phi} \\
&= \mathbf{d}^T \mathcal{L}^+ \mathbf{d} \quad \text{Since } \bar{\phi}^T \mathcal{L} \bar{\phi} = 1
\end{aligned} \tag{4.25}$$

Part 3

Let  $\mathbf{f}$  be any flow satisfying  $\Gamma^T \mathbf{f} = \mathbf{d}$ . Then we have:

$$\begin{aligned}
\mathcal{E}(\mathbf{P}, \mathbf{f}) &\geq (\mathbf{f}^T \mathbf{P} \mathbf{f}) (\bar{\phi}^T \Gamma^T \mathbf{P}^{-1} \Gamma \bar{\phi}) \quad \text{Since } \bar{\phi}^T \mathcal{L} \bar{\phi} = 1 \\
&= \|\mathbf{P}^{1/2} \mathbf{f}\|_2^2 \|\mathbf{P}^{-1/2} \Gamma \bar{\phi}\|_2^2 \\
&\geq (\mathbf{f}^T \Gamma \bar{\phi})^2 \quad \text{By Cauchy-Schwarz inequality} \\
&= (\mathbf{d}^T \bar{\phi})^2 \quad \text{Since } \mathbf{f} \text{ satisfies the demands} \\
&= \mathcal{E}(\mathbf{P}, \bar{\mathbf{f}}) \quad \text{By Part 2}
\end{aligned} \tag{4.26}$$

■

## 4.2 Finding Almost Optimal Vertex Potentials

The main part of computing an almost optimal set of vertex potentials from 4.21 is the computation of  $\mathcal{L}^+ \mathbf{d}$ . Since  $\mathbf{P}$  is no longer a diagonal, the matrix  $\mathcal{L}$  is no longer a Laplacian matrix. However, in certain more restrictive cases that still suffice for our purposes we can use lemma 2.1 of [BHV04]:

**Lemma 4.2** *For any matrices  $V, G, H$ , if  $H \preceq G \preceq \kappa H$ , then  $V H V^T \preceq G \preceq \kappa V H V^T$ .*

**Proof** Consider any vector  $x$ , we have:

$$\begin{aligned}
x^T V H V^T x &= (V^T x)^T H (V^T x) \\
&\preceq (V^T x)^T G (V^T x) = x^T V G V^T x
\end{aligned} \tag{4.27}$$

and

$$\begin{aligned}
x^T V G V^T x &= (V^T x)^T G (V^T x) \\
&\preceq \kappa (V^T x)^T H (V^T x) = \kappa x^T V H V^T x
\end{aligned} \tag{4.28}$$

■

This lemma allows us to precondition  $\mathbf{P}$  when each of  $\mathbf{P}(e)$  is well-conditioned, specifically:

**Lemma 4.3** *If there exist a constant  $\kappa$  such that for all  $e$ ,  $\kappa \lambda_{\min}(\mathbf{P}(e)) \geq \lambda_{\max}(\mathbf{P}(e))$ , then we can find a Laplacian matrix  $\tilde{\mathcal{L}}$  such that  $\tilde{\mathcal{L}} \preceq \mathcal{L} \preceq \kappa \tilde{\mathcal{L}}$ .*

**Proof** Consider replacing  $\mathbf{P}(e)$  with  $\tilde{\mathbf{P}}(e) = \lambda_{\min}(\mathbf{P}(e)) \mathbf{I}(e)$  where  $\mathbf{I}(e)$  is the  $k$  - by -  $k$  identity matrix. Then since  $\mathbf{P}(e) \preceq \lambda_{\max} \mathbf{I}(e)$  as well, we have:

$$\tilde{\mathbf{P}}(e) \preceq \mathbf{P}(e) \preceq \lambda_{\max}(\mathbf{P}(e))\mathbf{I}(e) = \frac{\lambda_{\max}(\mathbf{P}(e))}{\lambda_{\min}(\mathbf{P}(e))}\tilde{\mathbf{P}}(e) \quad (4.29)$$

Then applying Lemma 4.2 with  $V = \Gamma^T$ ,  $G = \mathbf{P}$  and  $H = \tilde{\mathbf{P}}$  gives:

$$\tilde{\mathcal{L}} \preceq \mathcal{L} \preceq \kappa \tilde{\mathcal{L}} \quad (4.30)$$

■

The following fact then allows us to solve linear equations on  $\mathcal{L}$  by solving linear systems on  $\tilde{\mathcal{L}}$  instead:

**Lemma 4.4** (*preconditioned Chebyshev*) [Saa96, Axe94] *Given matrix  $A$ , vector  $b$ , linear operator  $B$  and a constant  $\kappa$  such that  $B \preceq A^+ \preceq \kappa B$  and a desired error tolerance  $\theta$ . We can compute a vector  $x$  such that  $\|x - A^+b\|_A \leq \theta \|A^+b\|_A$  using  $O(\sqrt{\kappa} \log 1/\theta)$  evaluations of the linear operators  $A$  and  $B$ .*

Note that due to  $\Gamma$  being  $k$  copies of the edge-vertex incidence matrix, the matrix  $\tilde{L}$  is actually  $k$  Laplacians arranged in block-diagonal form. This allows us to apply SDD linear system solves to apply an operator that is close to the pseudo-inverse of  $\tilde{\mathcal{L}}$ , which we in turn use to solve systems involving  $\mathcal{L}$  using Lemma 4.4.

**Lemma 4.5** [ST06, KMP10, KMP11] *Given a Laplacian matrix of the form  $L = B^T W B$  for some diagonal matrix  $W \geq 0$ , there is a linear operator  $A$  such that*

$$A \preceq L^+ \preceq 2A$$

*And for any vector  $\mathbf{x}$ ,  $A\mathbf{x}$  can be evaluated in time  $\tilde{O}(m)$  where  $m$  is the number of non-zero entries in  $L$ .*

We can now prove the main result about solving systems involving  $\mathcal{L}$ .

**Lemma 4.6** *Given any set of energy matrices on edges  $\mathbf{P}$  such that  $\lambda_{\max}(\mathbf{P}(e)) \leq \kappa \lambda_{\min}(\mathbf{P}(e))$ , a vector  $\mathbf{d}$  and error parameter  $\delta$ . We can find an almost optimal set of vertex potentials  $\tilde{\mathbf{x}}$  such that:*

$$\|\tilde{\mathbf{x}} - \bar{\mathbf{x}}\|_{\mathcal{L}} \leq \delta \|\bar{\mathbf{x}}\|_{\mathcal{L}} \quad (4.31)$$

*In time  $\tilde{O}(mk^2 \sqrt{\kappa} \text{poly}(\epsilon^{-1}))$ .*

**Proof** Applying Lemma 4.5 to each of the Laplacians that make up  $\tilde{\mathcal{L}}$ , we can obtain a linear operator  $A$  such that:

$$\tilde{\mathcal{L}}^+ \preceq A \preceq 2\tilde{\mathcal{L}}^+ \quad (4.32)$$

Such that  $A\mathbf{x}$  can be evaluated in time  $\tilde{O}(mk)$ .

Combining these bounds then gives:

$$A \preceq 2\tilde{\mathcal{L}}^+ \preceq 2\kappa A \quad (4.33)$$

Then the running time follows from Lemma 4.4, which requires an extra  $\kappa$  iterations, and the fact that a forward multiply involving  $\mathbf{P}$  costs  $O(mk^2)$ . ■

Using this extension to the solver we can prove our main theorem about solving quadratically coupled flows, which we prove in Appendix A.

---

**Theorem 4.7** *There is an algorithm QUADRATICALLYCOUPLEDFLOW such that for any  $\delta > 0$  and  $F > 0$ , any set of energy matrices  $\mathbf{P}(e)$  such that  $I \preceq \mathbf{P}(e) \preceq UI$  for a parameter  $U$  and  $\kappa\lambda_{\min}(\mathbf{P}(e)) \geq \lambda_{\max}(\mathbf{P}(e))$ , and any demand vector  $\mathbf{d}$  such that the corresponding minimum energy flow is  $\tilde{\mathbf{f}}$ , computes in time  $\tilde{O}((\sqrt{\kappa}k^2 + k^\omega)m \log(U/\delta))$  a vector of vertex potentials  $\tilde{\phi}$  and a flow  $\tilde{\mathbf{f}}$  such that*

1.  $\tilde{\mathbf{f}}$  satisfies the demands in all the commodities  $\Gamma^T \tilde{\mathbf{f}} = \mathbf{d}$
2.  $\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P}) \leq (1 + \delta)\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P})$
3. for every edge  $e$ ,  $|\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P}, e) - \mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P}, e)| \leq \delta\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P})$
4. The energy given by the potentials  $\mathcal{E}(\mathbf{P}, \tilde{\phi})$  is at most 1. and its objective,  $\mathbf{d}^T \tilde{\phi}$  is at least  $(1 - \delta)\mathcal{C}_{eff}(\mathbf{P})$ .

## 5 Approximately Solving Quadratically Capacitated Flows

We now show that we can repeatedly solve quadratically coupled flows inside a multiplicative weights routine to minimize the maximum saturation of an edge. Pseudocode of our algorithm is shown in Algorithm 1.

The guarantees of this algorithm can be formalized as follows:

**Theorem 5.1** *Given a graph  $G = (V, E)$  and energy matrices  $\mathbf{P}(e)$  on each of the edges such that  $\kappa\lambda_{\min}\mathbf{P}(e) > \lambda_{\max}\mathbf{P}(e)$  and a parameter  $\epsilon$ , QUADRATICALLYCAPACITATEDFLOW returns one of the following in  $\tilde{O}(mk^2\kappa\epsilon^{-8/3})$  time:*

- A  $k$ -commodity flow  $\tilde{\mathbf{f}}$  such that:

$$\text{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}, e) \leq 1 + 10\epsilon$$

for all edges  $e$  and

$$\Gamma \tilde{\mathbf{f}} = \mathbf{d}$$

- **fail** indicating that there does not exist a  $k$ -commodity flow  $\mathbf{f}$  that satisfies all demands and have

$$\text{saturation}_{\mathbf{f}}(\mathbf{P}, e) \leq 1 - \epsilon$$

on all edges.

We first state the following bounds regarding the overall sum of potentials  $\mu^{(t)}$ , the weight of a single edge  $\mathbf{w}^{(t)}(e)$  and the effective conductance given by the reweighed energy matrices at each iteration,  $\mathcal{E}(\mathbf{P}^{(t)})$ .

**Lemma 5.2** *The following holds when  $\tilde{\mathbf{f}}$  satisfies*

$$\sum_e \text{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}^{(t-1)}, e) \leq \mu^{(t-1)}$$

- 1.

$$\mu^{(t)} \leq \exp\left(\frac{\epsilon}{\rho}\right) \mu^{(t-1)} \tag{5.34}$$

---

**Algorithm 1** Multiplicative weights update routine for approximately solving quadratically capacitated flows

---

QUADRATICALLYCAPACITATEDFLOW

Input: Weighted graph  $G = (V, E, w)$ , energy matrix  $\mathbf{P}(e)$  for each edge  $e$ , demands  $\mathbf{d}_1 \dots \mathbf{d}_k$  for each commodity. Error bound  $\epsilon$ .

Output: Either a collection of flows  $\mathbf{f}$  such that  $\text{saturation}_{\mathbf{f}}(\mathbf{P}, e) \leq 1 + \epsilon$ , or FAIL indicating that there does not exist a solution  $\mathbf{f}$  where  $\text{saturation}_{\mathbf{f}}(\mathbf{P}, e) \leq 1 - 2\epsilon$ .

```

1:  $\rho \leftarrow 10m^{1/3}\epsilon^{-2/3}$ 
2:  $N \leftarrow 20\rho \ln m\epsilon^{-2} = 200m^{1/3} \ln m\epsilon^{-8/3}$ 
3: Initialize  $\mathbf{w}^{(0)}(e) = 1$  for all  $e \in E$ 
4:  $\mathbf{f} \leftarrow \mathbf{0}$ 
5:  $N_1 \leftarrow 0$ 
6: for  $t = 1 \dots N$  do
7:   Compute  $\mu^{(t-1)} = \sum_e \mathbf{w}^{(t-1)}(e)$ 
8:   Compute reweighed energy matrices,  $\mathbf{P}^{(t-1)}(e) = (w^{(t-1)}(e) + \frac{\epsilon}{m}\mu^{(t-1)}) \mathbf{P}(e)$ 
9:   Query QUADRATICALLYCOUPLEDFLOW with energy matrices  $\mathbf{P}^{(t-1)}(e)$  and error bound  $\delta = \frac{\epsilon}{m}$ , let
       the flow returned be  $\tilde{\mathbf{f}}^{(t)}$ 
10:  if  $\mathcal{E}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}) > \mu^{(t-1)}$  then
11:    return fail
12:  else
13:    if  $\text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e) \leq \rho$  for all  $e$  then
14:       $\mathbf{f} \leftarrow \mathbf{f} + \tilde{\mathbf{f}}^{(t)}$ 
15:       $N_1 \leftarrow N_1 + 1$ 
16:    end if
17:    for  $e \in E$  do
18:       $\mathbf{w}^{(t)}(e) \leftarrow \mathbf{w}^{(t-1)}(e) \left(1 + \frac{\epsilon}{\rho} \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e)\right)$ 
19:    end for
20:  end if
21: end for
22: return  $\frac{1}{N_1} \mathbf{f}$ 

```

---

2.  $w^{(t)}(e)$  is non-decreasing in all iterations, and if  $\text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t)}, e) \leq \rho$ , we have:

$$w^{(t)}(e) \geq \exp\left(\frac{\epsilon}{\rho} \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t)}, e)\right) w^{(t-1)}(e) \quad (5.35)$$

3. If for some edge  $e$  we have  $\text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}, e) \geq \rho$ , then  $\mathcal{C}_{eff}(\mathbf{P}^{(t)}) \geq \mathcal{C}_{eff}(\mathbf{P}^{(t-1)}) \exp\left(\frac{\epsilon^2 \rho^2}{5m}\right)$

The proof of Lemma 5.2 relies on the following facts about  $\exp(x)$  when  $x$  is close to 1:

**Fact 5.3** 1. If  $x \geq 0$ ,  $1 + x \leq \exp(x)$ .

2. If  $0 \leq x \leq \epsilon$ , then  $1 + x \geq \exp((1 - \epsilon)x)$ .

**Proof of Part 1:**

$$\begin{aligned}
\mu^{(t)} &= \sum_e \mathbf{w}^{(t)}(e) \\
&= \sum_e \mathbf{w}^{(t-1)}(e) \left(1 + \frac{\epsilon}{\rho} \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}, e)\right) && \text{By the update rule} \\
&= \left(\sum_e \mathbf{w}^{(t-1)}(e)\right) + \frac{\epsilon}{\rho} \left(\sum_e \mathbf{w}^{(t-1)}(e) \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}, e)\right) \\
&\leq \mu^{(t-1)} + \frac{\epsilon}{\rho} \mu^{(t-1)} && \text{By definition of } \mu^{(t-1)} \text{ and total weighted saturation} \\
&= \left(1 + \frac{\epsilon}{\rho}\right) \mu^{(t-1)} \leq \exp\left(\frac{\epsilon}{\rho}\right) \mu^{(t-1)} && \text{By Fact 5.3.1}
\end{aligned} \tag{5.36}$$

■(Part 1)

**Proof of Part 2:**

If  $\text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t)}, e) \leq \rho$ , then  $\frac{\epsilon}{\rho} \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t)}, e) \leq \epsilon$  and:

$$\begin{aligned}
\mathbf{w}^{(t)}(e) &= \mathbf{w}^{(t-1)}(e) \left(1 + \frac{\epsilon}{\rho} \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t)}, e)\right) \\
&\leq \mathbf{w}^{(t-1)}(e) \exp\left(\frac{\epsilon(1-\epsilon)}{\rho} \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}, e)\right) && \text{By Fact 5.3.2}
\end{aligned} \tag{5.37}$$

■(Part 2)

**Proof of Part 3:**

Let  $e$  be the edge where  $\text{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}, e) \geq \rho$ , then since  $\mathbf{P}^{(t-1)}(e) \geq \frac{\epsilon}{m} \mu I$  by line 8, we have:

$$\begin{aligned}
\text{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}^{(t-1)}, e)^2 &\geq \frac{\epsilon}{3m} \mu^{(t-1)} \rho^2 \\
&\geq \frac{\epsilon \rho^2}{3m} \mathcal{E}(\mathbf{P}^{(t-1)}, \tilde{\mathbf{f}}) && \text{By assumption of the energy of the flow returned}
\end{aligned} \tag{5.38}$$

Invoking the guarantees proven in Theorem 4.7, we have:

$$\begin{aligned}
\text{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}^{(t-1)}, e)^2 &\geq \text{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}^{(t-1)}, e)^2 - |\text{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}^{(t-1)}, e)^2 - \text{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}^{(t-1)}, e)^2| \\
&\geq \text{saturation}_{\tilde{\mathbf{f}}}(\mathbf{P}^{(t-1)}, e)^2 - \delta \mathcal{E}(\mathbf{P}^{(t-1)}) && \text{By Part 3} \\
&\geq \frac{\epsilon \rho^2}{3m} \mathcal{E}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}) - \delta \mathcal{E}(\mathbf{P}^{(t-1)}) && \text{By Equation 5.38} \\
&\geq \frac{\epsilon \rho^2}{3(1+\delta)m} \mathcal{E}(\mathbf{P}^{(t-1)}) - \delta \mathcal{E}(\mathbf{P}^{(t-1)}) && \text{By Part 2} \\
&\geq \frac{\epsilon \rho^2}{4m} \mathcal{E}(\mathbf{P}^{(t-1)})
\end{aligned} \tag{5.39}$$

Then by the relation between  $\bar{\phi}$  and  $\tilde{\mathbf{f}}$ , we have that

$$\mathcal{E}_{\bar{\phi}^{(t-1)}}(\mathbf{P}^{(t-1)}(e)) \geq \frac{\epsilon \rho^2}{4m} \mathcal{E}_{\bar{\phi}^{(t-1)}}(\mathbf{P}^{(t-1)}) \tag{5.40}$$

Then since  $\mathbf{w}^{(t)}(e) \geq (1 + \epsilon)\mathbf{w}^{(t-1)}(e)$ , using the current set of optimal potential gives:

$$\mathcal{E}_{\bar{\phi}^{(t-1)}}(\mathbf{P}^{(t)}) \leq (1 - \frac{\epsilon^2 \rho^2}{4m}) \mathcal{E}(\mathbf{P}^{(t-1)}, \bar{\phi}) \quad (5.41)$$

Which means that when  $\epsilon < 0.01$ ,  $\sqrt{1 + \frac{\epsilon^2 \rho^2}{5m}} \bar{\phi}^{(t-1)}$  is a valid set of potentials for  $\mathbf{P}^{(t)}$  and therefore:

$$\begin{aligned} \mathcal{C}_{eff}(\mathbf{P}^{(t)}) &\geq (\mathbf{d}^T \sqrt{1 + \frac{\epsilon^2 \rho^2}{5m}} \bar{\phi}^{(t-1)})^2 \\ &\geq \exp\left(\frac{\epsilon^2 \rho^2}{5m}\right) (\mathbf{d}^T \bar{\phi}^{(t-1)})^2 \\ &= \exp\left(\frac{\epsilon^2 \rho^2}{5m}\right) \mathcal{C}_{eff}(\mathbf{P}^{(t-1)}) \end{aligned} \quad (5.42)$$

■(Part 3)

**Proof of Theorem 5.1:**

Since  $\sum_e (w^{(t-1)}(e) + \frac{\epsilon}{m} \mu^{(t-1)}) = (1 + \epsilon)\mu^{(t-1)}$ , if there exist a flow  $\mathbf{f}$  such that  $\text{saturation}_{\mathbf{f}}(\mathbf{P}, e) \leq 1 - 2\epsilon$  for all  $e$ , we have that  $\mathcal{E}_{\tilde{\mathbf{f}}^{(t)}}(\tilde{\mathbf{P}}) \leq (1 - \epsilon)\mu^{(t-1)}$ . Then if the algorithm does not return **fail**, Theorem 4.7 means that  $\tilde{\mathbf{f}}^{(t)}$  satisfies:

$$\begin{aligned} \mathcal{E}_{\tilde{\mathbf{f}}^{(t)}} &\leq (1 + \delta)(1 - \epsilon)\mu^{(t-1)} \\ &\leq \mu^{(t-1)} \end{aligned} \quad (5.43)$$

$$\sum_e \mathbf{w}^{(t-1)}(e) \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e)^2 \leq \sum_e \mathbf{w}^{(t-1)}(e) \quad (5.44)$$

Multiplying both sides by  $\mu^{(t-1)}$  and applying the Cauchy-Schwarz inequality gives:

$$\begin{aligned} \left(\sum_e \mathbf{w}^{(t-1)}(e)\right)^2 &\geq \left(\sum_e \mathbf{w}^{(t-1)}(e)\right) \left(\sum_e \mathbf{w}^{(t-1)}(e) \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e)^2\right) \\ &\geq \left(\sum_e \mathbf{w}^{(t-1)}(e) \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e)\right) \end{aligned} \quad (5.45)$$

Taking the square root of both sides gives:

$$\sum_e \mathbf{w}^{(t-1)}(e) \text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e) \leq \mu^{(t-1)} \quad (5.46)$$

Therefore inductively applying Lemma 5.2 Part1, we have:

$$\begin{aligned}
\mu^{(N)} &\leq \mu^{(0)} \cdot \left( \exp\left(\frac{\epsilon}{\rho}\right) \right)^N \\
&= \exp\left(\frac{\epsilon N}{\rho}\right) m \\
&\leq \exp\left(\frac{21 \ln m}{\epsilon}\right)
\end{aligned} \tag{5.47}$$

We now bound  $N'$ , the number of iterations  $t$  where there is an edge with  $\mathbf{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e) \geq \rho$ . Suppose  $\mathcal{C}_{eff}(\mathbf{P}^{(0)}) \leq 1/2$ , then in the flow returned, no edge  $e$  has  $\mathbf{saturation}_{\tilde{\mathbf{f}}^{(0)}}(\mathbf{P}^{(0)}, e) \geq 1$ , which means that the algorithm can already return that flow.

Then by the monotonicity of  $\mathcal{C}_{eff}(\mathbf{P}^{(t)})$  and Lemma 5.2 Part 3, we have:

$$\mathcal{C}_{eff}(\mathbf{P}^{(N)}) \geq 1/2 \cdot \exp\left(\frac{\epsilon^2 \rho^2 N'}{5m}\right) \tag{5.48}$$

Combining this with  $\mathcal{C}_{eff}(\mathbf{P}^{(t)}) \leq \mu^{(N)}$  gives:

$$\begin{aligned}
\frac{\epsilon^2 \rho^2 N'}{5m} &\leq \frac{21 \ln m}{\epsilon} \\
N' &\leq \frac{105m \ln m}{\rho^2 \epsilon^3} \\
&\leq \epsilon N
\end{aligned} \tag{5.49}$$

Then in all the  $N - N' \geq (1 - \epsilon)N$  iterations, we have  $\mathbf{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e) \leq \rho$  for all edges  $e$ . Then we have:

$$\begin{aligned}
\mathbf{saturation}_{\sum_t \tilde{\mathbf{f}}^{(t)}}(\mathbf{P}, e) &\leq \sum_t \mathbf{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}, e) \quad \text{Since } \mathbf{P}(e) \text{ defines a norm} \\
&\leq \log(\mu^{(t)}) / \left(\frac{\epsilon}{\rho}\right) \quad \text{By Lemma 5.2 Part 2} \\
&= \frac{1}{1 - \epsilon} T' \leq (1 + 2\epsilon) T'
\end{aligned} \tag{5.50}$$

■

## 6 Algorithm for Maximum Concurrent Multicommodity Flow

One of the main difficulties in directly applying the flow algorithm from [CKM<sup>+</sup>11] is that single commodity congestion constraints of the form  $\|\mathbf{f}(e)\|_1 \leq u_e$  are 'sharper' than the  $L_2$  energy functions due to the sign changes when each of the commodities are around 0.

As a result, we use the primal Primal-Dual SDP algorithm from [AK07] to generate the energy matrix. Pseudocode of the outermost layer of our algorithm for maximum concurrent flow is shown in Algorithm 2.

Where the update routine, UPDATE is shown in Algorithm 3. Note that  $\chi_i$  indicates the matrix that's 1 in entry  $(i, i)$  and 0 everywhere else, and  $\bar{\mathbf{M}}^{(t)}$  is used to store the sum of  $\mathbf{M}^{(r)}$  over  $1 \leq r \leq t$  to we do



---

**Algorithm 2** Algorithm for minimizing  $L_1$  congestion
 

---

MAXCONCURRENTFLOW

Input: Capacitated graph  $G = (V, E, \mathbf{u})$  and demands  $\mathbf{d}$ .

Algorithm for computing energy matrices based on a list of  $k$  flows, ENERGY and for minimizing the maximum energy along an edge. Iteration count  $N$  and error tolerance  $\epsilon$ .

Output: Either a flow  $\mathbf{f}$  that meets the demands and  $\|\mathbf{f}(e)\|_1 \leq (1 + 10\epsilon)\mu(e)$  on all edges, or **fail** indicating that there does not exist a flow  $\mathbf{f}$  that meets the demands and satisfy  $\|\mathbf{f}(e)\|_1 \leq \mu(e)$  on all edges.

```

1:  $\rho \leftarrow \sqrt{k\epsilon^{-1}}$ 
2:  $\epsilon_1 \leftarrow \frac{\epsilon}{k\rho} = \frac{\epsilon^{1/2}}{k^{3/2}}$ 
3:  $\epsilon'_1 \leftarrow -\ln(1 - \epsilon_1)$ 
4:  $N \leftarrow \rho\epsilon'_1{}^{-2} \log k = k^{7/2}\epsilon^{-3/2} \log k$ 
5: Initialize  $\bar{\mathbf{M}}^0(e) = \mathbf{0}$ ,  $\mathbf{W}^0(e) = \mathbf{I}$ 
6: for  $t = 1 \dots N$  do
7:   for  $e \in E$  do
8:      $\mathbf{P}^{(t-1)}(e) = \frac{1}{\mathbf{u}^2} \left( \mathbf{W}^{(t-1)}(e) / \|\mathbf{W}^{(t-1)}\|_\infty + \epsilon \mathbf{I} \right)$ 
9:   end for
10:  Query QUADRATICALLYCAPACITATEDFLOW with matrix  $\mathbf{P}^{(t-1)}$ 
11:  if QUADRATICALLYCAPACITATEDFLOW returns fail then
12:    return fail
13:  else
14:    Let the flow returned be  $\tilde{\mathbf{f}}^{(t)}$ 
15:    for  $e \in E$  do
16:       $(\bar{\mathbf{M}}^{(t)}(e), \mathbf{W}^{(t)}(e)) \leftarrow \text{UPDATE}(\bar{\mathbf{M}}^{(t-1)}(e), \mathbf{W}^{(t-1)}(e), \tilde{\mathbf{f}}^{(t)})$ 
17:    end for
18:  end if
19: end for
20: return  $\frac{1}{N} \sum_{t=1}^N \tilde{\mathbf{f}}^{(t)}$ 

```

---

not need to pass all of them to each invocation of UPDATE.

We start off by bounding the condition number of  $\mathbf{P}^{(t)}(e)$ .

**Lemma 6.1**

$$\lambda_{\max}(\mathbf{P}^{(t)}(e)) \leq 2k\epsilon^{-1}\lambda_{\min}(\mathbf{P}^{(t)}(e))$$

*Proof* Since  $\mathbf{W}(e)$  is positive semi definite, we have  $\mathbf{P}^{(t)}(e) \succeq \epsilon \mathbf{I}$ . Also, by definition of  $i^{(t)}(e)$ , we have that the maximum diagonal entry of  $\mathbf{W}^{(t-1)}(e) / \|\mathbf{W}^{(t-1)}(e)\|_\infty$  is 1. Therefore  $\text{tr}(\mathbf{P}^{(t)}(e)) \leq k + \text{tr}(\epsilon \mathbf{I}) \leq 2k$ . This in turn implies  $\lambda_{\max}(\mathbf{P}^{(t)}(e)) \leq 2k$ , which gives the required result.  $\blacksquare$

We first show that when a flow exist, QUADRATICALLYCAPACITATEDFLOW returns a flow  $\tilde{\mathbf{f}}^{(t)}(e)$  with low energy on each edge.

**Lemma 6.2** *If there is  $\bar{\mathbf{f}}$  such that  $\|\bar{\mathbf{f}}(e)\|_1 \leq 1$ , then at each iteration  $t$  we have  $\mathcal{E}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e) \leq 1 + 2\epsilon$ .*

*Proof*

---

**Algorithm 3** Matrix exponential based algorithm for generating energy matrix

---

UPDATE

Input:  $\bar{\mathbf{M}}^{(t-1)}(e)$ ,  $\mathbf{W}^{(t-1)}(e)$  from previous iterations, flow from iteration  $t$ ,  $\tilde{\mathbf{f}}^{(t)}(e)$ , Capacity  $\mathbf{u}(e)$ , Parameter  $\epsilon'_1$ .

Output: Sum matrix from current iteration  $S$ , Energy matrix  $X$ .

---

- 1: Find  $i^{(t-1)}(e) = \arg \max_i \mathbf{W}_{ii}^{(t-1)}(e)$
  - 2:  $\mathbf{M}^{(t)}(e) \leftarrow \frac{1}{2\rho} \left( (1 + 2\epsilon) \chi_{i^{(t-1)}(e)} - \frac{1}{\mathbf{u}(e)^2} \mathbf{f}^{(t)}(e) (\mathbf{f}^{(t)}(e))^T + \rho \mathbf{I} \right)$   
Where  $\chi_i$  has 1 in  $(i, i)$  and 0 everywhere else
  - 3:  $\bar{\mathbf{M}}^{(t)}(e) \leftarrow \bar{\mathbf{M}}^{(t-1)}(e) + \mathbf{M}^{(t)}(e)$
  - 4:  $\mathbf{W}^{(t)}(e) = \exp(-\epsilon'_1 \bar{\mathbf{M}}^{(t)}(e))$
  - 5: **return**  $(\bar{\mathbf{M}}^{(t)}(e), \mathbf{W}^{(t)}(e))$
- 

$$\begin{aligned}
\mathcal{E}_{\bar{\mathbf{f}}(e)}(\mathbf{W}^{(t-1)}(e)) &= \bar{\mathbf{f}}(e)^T \mathbf{W}^{(t-1)}(e) \bar{\mathbf{f}} \\
&= \sum_{ij} \bar{\mathbf{f}}_i(e) \bar{\mathbf{f}}_j(e) \mathbf{W}_{ij}^{(t-1)}(e) \\
&\leq \sum_{ij} |\bar{\mathbf{f}}_i(e)| |\bar{\mathbf{f}}_j(e)| |\mathbf{W}_{ij}^{(t-1)}(e)| \\
&\leq \|\mathbf{W}^{(t-1)}\|_\infty \left( \sum_{ij} |\bar{\mathbf{f}}_i(e)| |\bar{\mathbf{f}}_j(e)| \right) \\
&= \|\mathbf{W}^{(t-1)}\|_\infty \left( \sum_i |\bar{\mathbf{f}}_i(e)| \right)^2 \\
&\leq \|\mathbf{W}^{(t-1)}(e)\|_\infty \mathbf{u}(e)^2
\end{aligned} \tag{6.51}$$

Therefore we have:

$$\begin{aligned}
\mathcal{E}_{\bar{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e) &\leq \frac{1}{\mathbf{u}(e)^2} \left( \frac{1}{\|\mathbf{W}^{(t-1)}(e)\|_\infty} \mathcal{E}_{\bar{\mathbf{f}}(e)}(\mathbf{W}^{(t-1)}(e)) + \mathcal{E}_{\bar{\mathbf{f}}(e)}(\epsilon \mathbf{I}) \right) \\
&\leq 1 + 2\epsilon
\end{aligned} \tag{6.52}$$

Since  $\mathcal{E}_{\bar{\mathbf{f}}(e)}(\epsilon \mathbf{I}) = \epsilon \|\bar{\mathbf{f}}(e)\|_2^2 \leq \epsilon \|\bar{\mathbf{f}}(e)\|_1^2 \leq \mathbf{u}(e)^2$ .

And the properties of  $\mathbf{f}^{(t)}(e)$  follows from the guarantees of Theorem 5.1. ■

Using this width bound, we can now adapt the analysis in [AK07] to show that the sum of flows can be bounded in the matrix sense.

**Lemma 6.3** *If in all iterations the flows returned satisfy  $\mathbf{f}^{(t)}(e) (\mathbf{f}^{(t)}(e))^T \preceq \rho \mathbf{u}^2 \mathbf{I}$ , then we have the following by the end of  $N$  iterations:*

$$\left( \sum_{t=1}^N \frac{1}{\mathbf{u}(e)^2} \mathbf{f}^{(t)}(e) (\mathbf{f}^{(t)}(e))^T \right) \preceq (1 + 2\epsilon) \mathbf{u}^2 \sum_{t=0}^{N-1} \chi_{i(t)} + \frac{\epsilon N}{k} \mathbf{I}$$

**Proof**

The proof is similar to the proofs of Theorems 10 and 1 in Section 6 of [AK07]. We have:

$$\begin{aligned} \text{tr} \left( \mathbf{W}^{(t)}(e) \right) &= \text{tr} \left( \exp(-\epsilon'_1 \sum_{r=0}^t \mathbf{M}^{(r)}(e)) \right) \\ &\leq \text{tr} \left( \exp(-\epsilon'_1 \sum_{r=0}^{(t-1)} \mathbf{M}^{(r)}(e)) \exp(-\epsilon'_1 \mathbf{M}^{(t)}(e)) \right) && \text{By the Golden-Thompson inequality} \\ &= \text{tr} \left( \mathbf{W}^{(t-1)}(e) \exp(-\epsilon'_1 \mathbf{M}^{(t)}(e)) \right) \\ &\leq \text{tr} \left( \mathbf{W}^{(t-1)}(e) (\mathbf{I} - \epsilon'_1 \mathbf{M}^{(t)}(e)) \right) && \text{Since } \exp(-\epsilon'_1 \mathbf{A}) \preceq (\mathbf{I} - \epsilon'_1 \mathbf{A}) \text{ when } 0 \preceq \mathbf{A} \preceq \mathbf{I} \\ &= \text{tr}(\mathbf{W}^{(t-1)}(e)) - \epsilon'_1 \text{tr}(\mathbf{W}^{(t-1)}(e) \mathbf{M}^{(t)}(e)) \end{aligned} \quad (6.53)$$

The construction of  $\mathbf{W}(e)$  from line 4 of UPDATE means that  $\mathbf{W}(e)$  is positive semi-definite. This in turn implies that  $\|\mathbf{W}(e)\|_\infty \leq \max_i \mathbf{W}_{ii}$ . Substituting gives:

$$\begin{aligned} \text{tr} \left( \mathbf{W}^{(t-1)}(e) \mathbf{M}^{(t)}(e) \right) &= \text{tr} \left( \mathbf{W}^{(t-1)}(e) \cdot \left( (1 + 2\epsilon) \chi_{i(t-1)} - \mathbf{f}^{(t)}(e) (\mathbf{f}^{(t)}(e))^T + \rho \mathbf{I} \right) / 2\rho \right) \\ &= \frac{1}{2\rho} \left( (1 + 2\epsilon) \|\mathbf{W}^{(t-1)}(e)\|_\infty - \frac{1}{\mathbf{u}(e)^2} (\mathbf{f}^{(t)}(e))^T \mathbf{W}^{(t-1)}(e) \mathbf{f}^{(t)}(e) \right) + \frac{1}{2} \text{tr}(\mathbf{W}^{(t-1)}(e)) \\ &\leq \frac{1}{2\rho} \left( (1 + 2\epsilon) \|\mathbf{W}^{(t-1)}(e)\|_\infty - \|\mathbf{W}^{(t-1)}(e)\|_\infty (\mathbf{f}^{(t)}(e))^T \mathbf{P}^{(t-1)}(e) \mathbf{f}^{(t)}(e) \right) \\ &\quad + \frac{1}{2} \text{tr}(\mathbf{W}^{(t-1)}(e)) && \text{Since } \frac{\mathbf{W}^{(t-1)}(e)}{\mathbf{u}(e)^2 \|\mathbf{W}^{(t-1)}(e)\|_\infty} \preceq \mathbf{P}^{(t-1)}(e) \\ &\geq \frac{1}{2} \text{tr}(\mathbf{W}^{(t-1)}(e)) && \text{By Lemma 6.2} \end{aligned} \quad (6.54)$$

Combining these two gives:

$$\begin{aligned} \text{tr} \left( \exp(-\epsilon'_1 \bar{\mathbf{M}}^{(N)}(e)) \right) &= \text{tr}(\mathbf{W}^N(e)) \\ &\leq k \left( 1 - \frac{\epsilon_1}{2} \right)^N \\ &\leq k \exp \left( -\frac{\epsilon_1}{2} N \right) \\ &\leq \exp \left( -\frac{(1 - \epsilon_1) \epsilon_1}{2} N \right) && \text{Since } N = 10\rho\epsilon_1'^{-2} \log k \end{aligned} \quad (6.55)$$

Using the fact that  $\exp(-\lambda_{\max}(\mathbf{A})) \leq \text{tr}(-\exp(\mathbf{A}))$ , we get:

$$\begin{aligned} \frac{(1 - \epsilon_1)\epsilon_1}{2} N\mathbf{I} &\preceq \epsilon'_1 \sum_{t=0}^N \mathbf{M}^{(t)}(e) \\ &= \epsilon'_1 \sum_{t=0}^N \left( (1 + 2\epsilon)\chi_{i(t)} - \frac{1}{\mathbf{u}^2} \mathbf{f}^{(t)}(e)(\mathbf{f}^{(t)}(e))^T + \rho\mathbf{I} \right) / 2\rho \end{aligned} \quad (6.56)$$

$$-\epsilon'_1 \rho N\mathbf{I} \preceq \sum_{t=0}^N \left( (1 + 2\epsilon)\chi_{i(t)} - \frac{1}{\mathbf{u}^2} \mathbf{f}^{(t)}(e)(\mathbf{f}^{(t)}(e))^T \right) \quad (6.57)$$

$$\frac{1}{\mathbf{u}^2} \left( \sum_{t=1}^N \mathbf{f}^{(t)}(e)(\mathbf{f}^{(t)}(e))^T \right) \preceq (1 + 2\epsilon) \sum_{t=0}^{N-1} \chi_{i(t)} + \epsilon'_1 \rho N\mathbf{I} \quad (6.58)$$

Substituting in the setting of  $\epsilon_1 = \frac{\epsilon}{k\rho}$  gives the desired result. ■

This in turns lets us bound the  $L_1$  congestion of the flow returned after  $T$  iterations.

**Theorem 6.4** *After  $N = \tilde{O}(k^{7/2}\epsilon^{-5/2} \log k)$  iterations, MAXCONCURRENTFLOW returns a flow  $\mathbf{f}$  where for each edge  $e$ , we have:*

$$\left\| \frac{1}{N} \sum_{t=1}^N \mathbf{f}^{(t)} \right\|_1 \leq (1 + 3\epsilon) \mathbf{u}$$

**Proof**

We first bound the width of each update step. Note that by construction we have:

$$\begin{aligned} \frac{\epsilon}{\mathbf{u}^2 k} (\mathbf{f}^{(t)}(e))^T \mathbf{I} \mathbf{f}^{(t)}(e) &\leq (1 + \epsilon) \\ \frac{1}{\mathbf{u}^2} (\mathbf{f}^{(t)}(e))^T \mathbf{I} \mathbf{f}^{(t)}(e) &\leq 2k\epsilon^{-1} \end{aligned} \quad (6.59)$$

Then by the Cauchy-Schwarz inequality we have:

$$\begin{aligned} \left\| \frac{\mathbf{f}^{(t)}(e)}{\mathbf{u}(e)} \right\|_1^2 &\leq k \left\| \frac{\mathbf{f}^{(t)}(e)}{\mathbf{u}(e)} \right\|_2^2 \\ &= 2k\epsilon^{-1} \end{aligned} \quad (6.60)$$

Which gives that  $\rho = \sqrt{2k\epsilon^{-1}}$  suffices as width parameter.

Let  $\mathbf{s}$  be the vector corresponding to the signs of the entries of  $\sum_{t=1}^N \mathbf{f}^{(t)}(e)$ , aka.  $\mathbf{s}^T \sum_{t=1}^N \mathbf{f}^{(t)}(e) = \|\sum_{t=1}^N \mathbf{f}^{(t)}(e)\|_1$ . Then:

$$\begin{aligned}
\left\| \frac{1}{\mathbf{u}N} \sum_{1 \leq t \leq N} \mathbf{f}^{(t)}(e) \right\|_1^2 &= \left( \mathbf{s}^T \frac{1}{N} \sum_{t=1}^N \mathbf{f}^{(t)}(e) \right)^2 \\
&= \frac{1}{\mathbf{u}^2 N^2} \left( \mathbf{s}^T \sum_{t=1}^N \mathbf{f}^{(t)}(e) \right)^2 \\
&\leq \frac{1}{N} \sum_{t=1}^N \left( \mathbf{s}^T \mathbf{f}^{(t)}(e) \right)^2 \quad \text{By the Cauchy-Schwarz inequality} \\
&= \frac{1}{N} \mathbf{s}^T \left( \sum_{t=1}^N \frac{1}{\mathbf{u}^2} \mathbf{f}^{(t)}(e) (\mathbf{f}^{(t)}(e))^T \right) \mathbf{s} \\
&\leq \frac{1}{N} \mathbf{s}^T \left( \sum_{t=1}^N (1 + 2\epsilon) \chi_{i(t)} + \frac{\epsilon N}{k} \mathbf{I} \right) \mathbf{s} \quad \text{By Lemma 6.3} \\
&= 1 + 3\epsilon \quad \text{Since } \mathbf{s}_i = \pm 1 \text{ and } \mathbf{s}^T \mathbf{s} = k
\end{aligned} \tag{6.61}$$

■

The running time of the algorithm can then be bounded as follows:

**Lemma 6.5** *Each iteration of the MAXCONCURRENTFLOW runs in  $\tilde{O}(m^{4/3}k^{5/2}\epsilon^{-19/6} + mk^\omega)$  time, giving an overall running time of  $\tilde{O}(m^{4/3}k^6\epsilon^{-17/3} + mk^{7/2+\omega}\epsilon^{-5/2})$ , where  $\omega$  is the matrix multiplication exponent.*

**Proof** The first term follows from the running time of QUADRATICALLYCAPACITATEDFLOW proven in Theorem 5.1 and  $\kappa(\mathbf{P}(e)) \leq O(\sqrt{k\epsilon^{-1}})$  from Lemma 6.1. For the second term, the bottleneck is the computation of matrix exponentials. This can be done in **poly**( $\log k$ ) matrix multiplies using [YL93], giving the  $\tilde{O}(k^\omega)$  bound in the each of the iterations. ■

## 7 Alternative Outer Algorithm

We show a modified formulation of the capacity bounds as  $2^k$  constraints per edge that brings us back to minimizing the maximum congestion. This gives a more combinatorial approach to minimizing the maximum  $L_1$  congestion, although the algorithm is slightly more intricate. As the computation of energy matrices only rely on the sum of flows so far (aka. history independent), we describe its computation in a separate routine ENERGY and first state the overall algorithm in Algorithm 4.

We start with the following observation that the maximum among the sums given by all  $2^k$  choices of signs to  $\mathbf{f}_i(e)$  equals congestion.

### Observation 7.1

$$\sum_{i=1}^k |\mathbf{f}_i(e)| = \max_{s_1, s_2, \dots, s_k \in \{-1, 1\}^k} \sum_i s_i \mathbf{f}_i(e)$$

We let  $\mathcal{S}$  to denote the set of all  $2^k$  settings of signs. This allows us to reformulate the constraint of  $\|\mathbf{f}(e)\|_1 \leq \mathbf{u}(e)$  as:

$$\mathbf{s}^T \mathbf{f}(e) \leq \mathbf{u}(e) \quad \forall \mathbf{s} \in \mathcal{S} \tag{7.62}$$

---

**Algorithm 4** Alternate Algorithm for Maximum Concurrent Multicommodity Flow

---

MAXCONCURRENTFLOW1

Input: Capacitated graph  $G = (V, E, \mathbf{u})$  and demands  $\mathbf{d}$ .

Algorithm for computing energy matrices based on a list of  $k$  flows, ENERGY and for minimizing the maximum energy along an edge, QUADRATICALLYCAPACITATEDFLOW. Width parameter  $\rho$ , iteration count  $N$  and error tolerance  $\epsilon$ .

Output: Either a flow  $\tilde{\mathbf{f}}$  that meets the demands and  $\|\tilde{\mathbf{f}}(e)\|_1 \leq (1 + 10\epsilon)\mu(e)$  on all edges, or **fail** indicating that there does not exist a flow  $\mathbf{f}$  that meets the demands and satisfy  $\|\mathbf{f}(e)\|_1 \leq \mu(e)$  on all edges.

```

1: for  $t = 1 \dots N$  do
2:   for  $e \in E$  do
3:      $\mathbf{P}^{(t)}(e) = \text{ENERGY}(\sum_{1 \leq r < t} \mathbf{f}^{(r)}(e))$ 
4:   end for
5:   Query QUADRATICALLYCAPACITATEDFLOW with matrix  $\mathbf{P}^{(t)}$ 
6:   if QUADRATICALLYCAPACITATEDFLOW returns fail then
7:     return fail
8:   end if
9: end for
10: return  $\frac{1}{N} \sum_{t=1}^N \mathbf{f}^{(N)}$ 

```

---

This reduces the problem back to minimizing the maximum among all  $|\mathcal{S}| = 2^k$  dot products with  $\mathbf{f}(e)$ . To solve this problem we can once again apply the multiplicative weights framework. We state the convergence result in a more general form:

**Theorem 7.2** *If for all flows  $\mathbf{f}(e)$ ,  $\text{ENERGY}(\mathbf{f}(e))$  returns a matrix*

$$\mathbf{P}(e) = \frac{1}{\sum_{s \in \mathcal{S}} \tilde{\mathbf{w}}(s)} \sum_{s \in \mathcal{S}} \frac{\tilde{\mathbf{w}}(e)}{\mathbf{u}(e)^2} \mathbf{s} \mathbf{s}^T + \frac{\epsilon}{\mathbf{u}(e)^2} \mathbf{I} \quad (7.63)$$

Where  $\tilde{\mathbf{w}}(s)$  satisfies

$$\exp\left(\frac{\epsilon \mathbf{s}^T \mathbf{f}(e)}{\rho \mathbf{u}(e)}\right) \leq \tilde{\mathbf{w}}(s) \leq (1 + \epsilon) \exp\left(\frac{\epsilon \mathbf{s}^T \mathbf{f}(e)}{\rho \mathbf{u}(e)}\right) \quad (7.64)$$

Then:

1.  $\lambda_{\max}(\mathbf{P}(e)) \leq 2k\epsilon^{-1}\lambda_{\min}(\mathbf{P}(e))$ .
2. If there exist a flow  $\tilde{\mathbf{f}}$  that meets all the demands and have  $\|\tilde{\mathbf{f}}(e)\|_1 \leq (1 - 3\epsilon)\mathbf{u}(e)$ , MAXCONCURRENTFLOW with  $\rho = k$  and  $N = \rho k \epsilon^{-2}$  returns a flow  $\tilde{\mathbf{f}}$  that meets the demands and satisfy  $\|\tilde{\mathbf{f}}(e)\|_1 \leq (1 + 3\epsilon)\mathbf{u}(e)$  over all edges.

**Proof of Part 1:**

For  $\lambda_{\min}(\mathbf{P}(e))$ , we have  $\mathbf{P}(e) \succeq \frac{\epsilon}{\mathbf{u}(e)^2} \mathbf{I}$  and the condition number bound follows from  $\lambda_{\min}(\mathbf{I}) = 1$ .

We can bound the maximum eigenvalue with the trace. Note that  $\text{tr}(\mathbf{s} \mathbf{s}^T) = k$  since the diagonal of  $\mathbf{s} \mathbf{s}^T$  is all 1. This gives:

---


$$\begin{aligned}
\lambda_{\max}(\mathbf{P}(e)) &\leq \text{tr}(\mathbf{P}(e)) \\
&= \frac{1}{\sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})} \sum_{\mathbf{s} \in \mathcal{S}} \frac{\tilde{\mathbf{w}}(e)}{\mathbf{u}(e)^2} \text{tr} \mathbf{s} \mathbf{s}^T + \frac{\epsilon}{\mathbf{u}(e)^2} \text{tr} \mathbf{I} \\
&= \frac{1 + \epsilon}{\mathbf{u}(e)^2} k
\end{aligned} \tag{7.65}$$

Therefore we get:

$$\begin{aligned}
\frac{\lambda_{\max}(\mathbf{P}(e))}{\lambda_{\min}(\mathbf{P}(e))} &\leq \frac{1 + \epsilon}{\mathbf{u}(e)^2} k \epsilon^{-1} \\
&\leq 2k \epsilon^{-1}
\end{aligned} \tag{7.66}$$

■

**Proof of Part 2:**

We define the exact set of weights that  $\tilde{\mathbf{w}}(\mathbf{s})$  are trying to approximate:

$$\mathbf{w}(\mathbf{s})^{(t)} = \exp \left( \frac{\epsilon \mathbf{s}^T \mathbf{f}^{(t)}}{\rho \mathbf{u}(e)} \right) \tag{7.67}$$

Also, let  $\bar{\mathbf{w}}^{(t)} = \sum_{\mathbf{s} \in \mathcal{S}} \mathbf{w}(\mathbf{s})^{(t)}$ . We have that at any iteration:

$$\begin{aligned}
\bar{\mathbf{w}}^{(t)} &\geq \max_{\mathbf{s}} \mathbf{w}(\mathbf{s})^{(t)} \\
&= \exp(\max_{\mathbf{s}} \mathbf{f}^{(t)}) \\
&= \exp(\|\mathbf{f}^{(t)}\|_1)
\end{aligned} \tag{7.68}$$

Therefore it suffices to upper bound the value of  $\mu^{(t)}$ . First note that if there is a flow  $\bar{\mathbf{f}}$  such that  $\|\bar{\mathbf{f}}(e)\|_1 \leq (1 - 3\epsilon)\mathbf{u}(e)$ , then we have  $\mathbf{s}^T \bar{\mathbf{f}}(e) \leq \|\bar{\mathbf{f}}(e)\|_1 \leq \mathbf{u}(e)$  for any  $\mathbf{s}$ . Squaring this and taking sum over all  $\mathbf{s} \in \mathcal{S}$  gives:

$$\begin{aligned}
\bar{\mathbf{f}}(e)^T \mathbf{P}(e)^{(t-1)} \bar{\mathbf{f}}(e) &= \bar{\mathbf{f}}(e)^T \left( \frac{1}{\sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)}} \sum_{\mathbf{s} \in \mathcal{S}} \frac{\tilde{\mathbf{w}}(e)^{(t-1)}}{\mathbf{u}(e)^2} \mathbf{s} \mathbf{s}^T + \frac{\epsilon}{k \mathbf{u}(e)^2} \mathbf{I} \right) \bar{\mathbf{f}}(e) \\
&= \frac{1}{\sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)}} \sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)} \left\| \frac{\mathbf{s}^T \bar{\mathbf{f}}(e)}{\mathbf{u}(e)} \right\|^2 + \epsilon \left\| \frac{\bar{\mathbf{f}}(e)}{\mathbf{u}(e)} \right\|_2^2 \\
&\leq \frac{1}{\sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)}} (1 - 3\epsilon)^2 \sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)} + \epsilon \\
&\leq (1 - 2\epsilon)^2
\end{aligned} \tag{7.69}$$

This means that there exist a flow  $\mathbf{f}$  such that:

---


$$\text{saturation}_{\mathbf{f}}(\mathbf{P}^{(t-1)}, e) \leq 1 - 2\epsilon$$

Therefore by Theorem 5.1, we have that for all edges  $e$ :

$$\text{saturation}_{\tilde{\mathbf{f}}^{(t)}}(\mathbf{P}^{(t-1)}, e) \leq 1 - \epsilon$$

This has two consequences:

1.

$$\begin{aligned} \frac{\epsilon}{\mathbf{u}(e)^2} (\tilde{\mathbf{f}}^{(t)})^T \mathbf{I} \tilde{\mathbf{f}}^{(t)} &\leq 1 - \epsilon \quad \text{Since } \mathbf{P}(e) \succeq \frac{\epsilon}{\mathbf{u}(e)^2} \mathbf{I} \\ \left\| \frac{\tilde{\mathbf{f}}^{(t)}}{\mathbf{u}(e)} \right\|_1^2 &\leq k \left\| \frac{\tilde{\mathbf{f}}^{(t)}}{\mathbf{u}(e)} \right\|_2^2 = \frac{k}{\epsilon} \quad \text{By the Cauchy-Schwarz inequality} \end{aligned} \quad (7.70)$$

Squaring both sides gives that  $\frac{\|\tilde{\mathbf{f}}^{(t)}\|_1}{\mathbf{u}(e)} \leq \sqrt{k}\epsilon^{-1/2}$ , which allows us to bound the width of the multiplicative updates.

2. Expanding out the first term in the formulation of  $\mathbf{P}(e)$  gives:

$$\sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)} \left( \frac{\mathbf{s}^T \tilde{\mathbf{w}}(e)}{\mathbf{u}(e)} \right)^2 \leq (1 - \epsilon) \sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)} \quad (7.71)$$

Multiplying both sides by  $\sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)}$  and applying Cauchy-Schwarz inequality gives:

$$\sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)} \left| \frac{\mathbf{s}^T \tilde{\mathbf{w}}(e)}{\mathbf{u}(e)} \right| \leq \sum_{\mathbf{s} \in \mathcal{S}} \tilde{\mathbf{w}}(\mathbf{s})^{(t-1)} \quad (7.72)$$

Combining with the fact that  $\mathbf{w} \leq \tilde{\mathbf{w}} \leq (1 + \epsilon)\mathbf{w}$  gives:

$$\sum_{\mathbf{s} \in \mathcal{S}} \mathbf{w}(\mathbf{s})^{(t-1)} \left| \frac{\mathbf{s}^T \tilde{\mathbf{w}}(e)}{\mathbf{u}(e)} \right| \leq (1 + \epsilon) \sum_{\mathbf{s} \in \mathcal{S}} \mathbf{w}(\mathbf{s})^{(t-1)} \quad (7.73)$$

Using Fact 5.3 we have:



---


$$\begin{aligned}
\bar{\mathbf{w}}^{(t-1)} &= \sum_{\mathbf{s} \in \mathcal{S}} \mathbf{w}(\mathbf{s})^{(t-1)} \exp \left( \frac{\epsilon \mathbf{s}^T \tilde{\mathbf{w}}(e)}{\rho \mathbf{u}(e)} \right) \\
&\leq \sum_{\mathbf{s} \in \mathcal{S}} \mathbf{w}(\mathbf{s})^{(t-1)} \exp \left( \frac{\epsilon}{\rho} \left| \frac{\mathbf{s}^T \tilde{\mathbf{w}}(e)}{\mathbf{u}(e)} \right| \right) && \text{Since } \exp(x) \text{ is monotonic and } x \leq |x| \\
&\leq \sum_{\mathbf{s} \in \mathcal{S}} \mathbf{w}(\mathbf{s})^{(t-1)} \frac{(1+2\epsilon)\epsilon}{\rho} \left| \frac{\mathbf{s}^T \tilde{\mathbf{w}}(e)}{\mathbf{u}(e)} \right| && \text{By Fact 5.3 Part 2 and } \left| \frac{\mathbf{s}^T \tilde{\mathbf{w}}(e)}{\mathbf{u}(e)} \right| \leq \rho \\
&\leq \frac{\epsilon}{\rho} (1+2\epsilon)(1+\epsilon) \sum_{\mathbf{s} \in \mathcal{S}} \mathbf{w}(\mathbf{s})^{(t-1)} \\
&\leq \frac{\epsilon(1+4\epsilon)}{\rho} \bar{\mathbf{w}}^{(t-1)} \\
&\leq \exp \left( \frac{\epsilon(1+4\epsilon)}{\rho} \right) \bar{\mathbf{w}}^{(t-1)} && \text{By Fact 5.3 Part 1}
\end{aligned} \tag{7.74}$$

Applying this inductively along with the fact that  $\bar{\mathbf{w}}^{(0)} = 2^k$  gives:

$$\begin{aligned}
\bar{\mathbf{w}}^{(t)} &\leq 2^k \exp \left( \frac{\epsilon(1+4\epsilon)}{\rho} \right)^t \\
&= \exp \left( \frac{\epsilon(1+4\epsilon)t}{\rho} + k \right)
\end{aligned} \tag{7.75}$$

Substituting in  $N = \frac{\rho k}{\epsilon^2}$  gives:

$$\begin{aligned}
\bar{\mathbf{w}}^N &\leq \exp \left( \frac{\epsilon(1+4\epsilon)\rho k}{\rho \epsilon^2} + k \right) \\
&= \exp \left( \frac{(1+5\epsilon)k}{\epsilon} \right) \\
&= \exp \left( \frac{\epsilon}{\rho} (1+5\epsilon) N \right)
\end{aligned} \tag{7.76}$$

Which gives  $\frac{1}{N} |\mathbf{f}^{(N)}(e)|_1 \leq (1+\epsilon) \mathbf{u}(e)$ . ■

## 7.1 Efficient Estimation of the Energy Matrix

The algorithm as stated has an iteration complexity that's  $O(k^{3/2} \epsilon^{-5/2})$ , which is small enough for our purposes.

However, a direct implementation of the generation of the energy matrix  $\mathbf{P}(e)$  requires looping through each of the  $2^k$  sign vectors  $\mathbf{s} \in \mathcal{S}$  and computing  $\mathbf{s}^T \mathbf{f}$ , which takes time exponential in  $k$ .

To alleviate this problem, note that the requirement of Theorem 7.2 allows us to compute the matrix for some set of weights  $\tilde{\mathbf{w}}$  where  $\mathbf{w}(\mathbf{s}) \leq \tilde{\mathbf{w}} \leq (1+\epsilon) \mathbf{w}(\mathbf{s})$ . Specifically we show the following:

**Theorem 7.3** *Given a flow  $\mathbf{f}$  such that  $\|\mathbf{f}\|_1 \leq \rho' \mathbf{u}$ , there is an algorithm ENERGY that computes a matrix  $\mathbf{P}$  where*

$$P = \sum_s \tilde{w} s s^T \quad (7.77)$$

$$\exp\left(\frac{s^T \mathbf{f}}{\mathbf{u}}\right) \leq \tilde{w} \leq (1 + \epsilon) \exp\left(\frac{s^T \mathbf{f}}{\mathbf{u}}\right) \quad (7.78)$$

In  $\tilde{O}(k^4 \rho' \epsilon^{-1})$  time.

Since we have the signs of each of the  $\mathbf{f}_i$ , we can easily find the value  $\mathbf{s}$  that maximizes  $\mathbf{s}^T \mathbf{f}$ . We let this set of signs be  $\bar{\mathbf{s}}$ , then we have for all  $\mathbf{s} \in \mathcal{S}$ :

$$\exp\left(\frac{\mathbf{s}^T \mathbf{f}}{\mathbf{u}}\right) = \exp\left(\frac{\bar{\mathbf{s}}^T \mathbf{f}}{\mathbf{u}}\right) \exp\left(-\frac{(\bar{\mathbf{s}} - \mathbf{s})^T \mathbf{f}}{\mathbf{u}}\right) \quad (7.79)$$

The first term is a constant, therefore it suffices to get good approximations for the second term. To do so we round each entry of  $\mathbf{f}$  to the lowest integral multiple of  $\frac{\epsilon}{k}$  and bound the error as follows:

**Lemma 7.4** *Let  $\tilde{\mathbf{f}}$  be  $\mathbf{f}$  with each entry rounded towards 0 to the nearest multiple of  $\frac{\epsilon}{3k} \mathbf{u}$ . Then we have:*

$$\begin{aligned} \exp\left(-\frac{(\bar{\mathbf{s}} - \mathbf{s})^T \mathbf{f}}{\mathbf{u}}\right) &\leq \exp\left(-\frac{(\bar{\mathbf{s}} - \mathbf{s})^T \tilde{\mathbf{f}}}{\mathbf{u}}\right) \\ &\leq (1 + \epsilon) \exp\left(-\frac{(\bar{\mathbf{s}} - \mathbf{s})^T \tilde{\mathbf{f}}}{\mathbf{u}}\right) \end{aligned} \quad (7.80)$$

Also,  $\|\tilde{\mathbf{f}}\|_1 \leq \rho' \mathbf{u}$  as well.

**Proof of Theorem 7.3:** Note that by the choice of  $\bar{\mathbf{s}}$ ,  $\bar{\mathbf{s}}_i \mathbf{f}_i \geq 0$  in each of commodity  $i$ . Therefore  $(\bar{\mathbf{s}} - \mathbf{s})_i \mathbf{f}_i$  is either 0 or  $2\mathbf{f}_i$ . By the rounding rule we have:

$$|\mathbf{f}_i| - \frac{\epsilon}{3k} \mathbf{u} \leq |\tilde{\mathbf{f}}_i| \leq |\mathbf{f}_i| \quad (7.81)$$

Which gives us the bound on  $\|\tilde{\mathbf{f}}\|_1$ . When combined with the fact that  $\tilde{\mathbf{f}}_i$  having the same sign as  $\mathbf{f}_i$  gives:

$$(\bar{\mathbf{s}} - \mathbf{s})_i \mathbf{f}_i - \frac{2\epsilon}{3k} \mathbf{u} \leq (\bar{\mathbf{s}} - \mathbf{s})_i \tilde{\mathbf{f}}_i \leq (\bar{\mathbf{s}} - \mathbf{s})_i \mathbf{f}_i \quad (7.82)$$

Summing this over the  $k$  commodities gives:

$$(\bar{\mathbf{s}} - \mathbf{s})^T \mathbf{f} - \frac{2\epsilon}{3} \mathbf{u} \leq (\bar{\mathbf{s}} - \mathbf{s})^T \tilde{\mathbf{f}} \leq (\bar{\mathbf{s}} - \mathbf{s})^T \mathbf{f} \quad (7.83)$$

Exponentiating both sides of Fact 5.3 Part 2 gives  $\exp(\frac{2\epsilon}{3}) \leq (1 + \epsilon)$ , from which the result follows. ■

After this rounding, the values of  $\mathbf{s}^T \mathbf{f}$  can only be multiples of  $\frac{\epsilon}{k} \mathbf{u}$  between  $[-\rho' \mathbf{u}, \rho' \mathbf{u}]$ . This allows us to narrow down the number of possible values of  $\exp(\frac{\mathbf{s}^T \mathbf{f}}{\mathbf{u}})$  to one of  $O(\rho' k \epsilon - 1)$  values. Further more,

notice that to calculate  $\mathbf{P}_{ij}$  it suffices to find the list of values of  $\frac{\mathbf{s}^T \mathbf{f}}{\mathbf{u}}$  among all  $\mathbf{s}$  such that  $\mathbf{s}_i = \mathbf{s}_j$  and the list where  $\mathbf{s}_i \neq \mathbf{s}_j$ . Each of these calculations can be done in  $\tilde{O}(\rho' k^2 \epsilon^{-1})$  time using the following lemma:

**Lemma 7.5** *Given a list of positive integer values  $a_1, a_2, \dots, a_k$  such that  $\sum_i a_i = N$ , there is an algorithm CONVOLVEALL that computes in  $\tilde{O}(Nk \log^2 N)$  time for each  $j \in [1, N]$  the number of subsets  $S \subseteq \{1, 2, \dots, k\}$  such that  $\sum_{i \in S} a_i = j$ .*

**Proof** Since the ordering is irrelevant, we may assume that  $a_1 \leq a_2 \leq \dots \leq a_k$ . Then there exist an index  $i$  such that  $\sum_{1 \leq j \leq i} a_j$  and  $\sum_{i+1 \leq j \leq k-1} a_j$  are both at most  $N/2$ . Suppose we have two lists containing the number of sums for each value between 0 and  $N/2$ , then taking their convolution can be done in  $O(N \log N)$  multiplications involving  $k$  bit numbers ([CSRL01] chapter 30). The last entry of  $a_k$  can be incorporated similarly. This leads us to the following recurrence on  $T(N)$ , the time required to compute the answer when the total sum is  $N$ :

$$T(N) \leq 2T(N/2) + \tilde{O}(Nk \log N) \quad (7.84)$$

Solving gives  $T(N) = \tilde{O}(Nk \log^2 N)$ . ■

---

**Algorithm 5** Algorithm for computing approximate energy matrix

---

ENERGY

Input: A  $k$  commodity flow  $\mathbf{f}$ . Capacity  $\mathbf{u}$ , parameter  $\rho'$  such that  $\|\mathbf{f}\|_1 \leq \rho' \mathbf{u}$ . Error bound  $\epsilon$ .

Output: Approximate energy matrix satisfying the guarantees of Theorem 7.3

---

```

1: Compute the set of signs that maximizes  $\mathbf{s}^T \mathbf{f}$ ,  $\bar{\mathbf{s}}$ 
2: for  $i = 1 \dots k$  do
3:   if  $\mathbf{f}_i \geq 0$  then
4:      $\tilde{\mathbf{f}}_i = \mathbf{u} \frac{\epsilon}{k} \lfloor \frac{\mathbf{f}_i}{k} \epsilon^{-1} \rfloor$ 
5:   else
6:      $\tilde{\mathbf{f}}_i = -\mathbf{u} \frac{\epsilon}{k} \lfloor -\frac{\mathbf{f}_i}{k} \epsilon^{-1} \rfloor$ 
7:   end if
8: end for
9: for  $i = 1 \dots k$  do
10:  for  $j = 1 \dots k$  do
11:    for Each setting of  $\mathbf{s}_i, \mathbf{s}_j$  do
12:      For each  $l \neq i, j$ , create  $a_l = 2 \frac{\tilde{\mathbf{f}}_i}{\mathbf{u}} k \epsilon^{-1}$ 
13:       $\mathbf{b} \leftarrow \text{CONVOLVEALL}(\mathbf{a})$ 
14:      for  $-\rho' k \epsilon^{-1} \leq l \leq \rho' k \epsilon^{-1}$  do
15:         $\mathbf{P}_{ij} \leftarrow \mathbf{P}_{ij} + \mathbf{b}_l \cdot \mathbf{s}_i \cdot \mathbf{s}_j \cdot \exp(\frac{\epsilon}{\rho} \frac{\bar{\mathbf{s}}^T \mathbf{f}}{\mathbf{u}} - l)$ 
16:      end for
17:    end for
18:  end for
19: end for
20: return  $\mathbf{P}$ 
```

---

The overall pseudocode for computing this energy matrix is shown in Algorithm 5. Summing over all  $O(k^2)$  entries gives the total running time. It's worth noting that because the matrix entries consists of differences of weights, the matrix that we obtain can have some entries that are very different than what

we would obtain if we use the exact values of  $\mathbf{w}(\mathbf{s})$ . Their similarity is obtained through the similarity of  $\tilde{w}$  and  $\hat{w}$  as they are weights on positive semi-definite outer products.

We can now bound the overall running time of the algorithm:

**Corollary 7.6** `MAXCONCURRENTFLOW1` runs in  $\tilde{O}(k^{3/2}\epsilon^{-5/2})$  iterations, where each iteration takes time  $\tilde{O}(m^{4/3}k^{5/2}\epsilon^{-19/6} + mk^5\epsilon^{-3/2})$ , for a total running time of  $\tilde{O}(m^{4/3}k^4\epsilon^{-17/3} + mk^{13/2}\epsilon^{-4})$ .

**Proof** The iteration count from 7.2 completes the proof. The first term follows from Theorem 5.1 and  $\kappa(\mathbf{P}(e)) \leq \sqrt{k}\epsilon^{-1/2}$ , Theorem 7.2 Part 2 gives that  $\|\mathbf{f}^{(t)}\|_1 \leq O(N)$ , which gives  $\|\epsilon \mathbf{f}^{(t)}\|_1 \leq \frac{\epsilon N}{\rho} = \tilde{O}(k)$ . Letting  $\rho' = \tilde{O}(k)$  in Theorem 7.3 then gives the second term in the bound. ■

## 8 Comments/Extensions

We have shown an approach of dealing with the coupling of the  $k$  commodities by associating an energy matrix with them. This allows us to approximate multicommodity flows in time  $\tilde{O}(m^{4/3}\text{poly}(k, \epsilon^{-1}))$ . We believe that our approach is quite general and extends naturally to other couplings between sets of  $k$  flows/vertex labels, with the most natural generalization being Markov random fields [KS80, SZS<sup>+</sup>08]. Since reductions from multicommodity flow to  $\tilde{O}(k\epsilon^{-2})$  calls of minimum cost flows are known [LMP<sup>+</sup>91]. A stronger result would be an approximation of minimum cost flow that runs in  $\tilde{O}(m^{4/3}\text{poly}(k, \epsilon^{-1}))$  time. This problem is significantly harder due to it incorporating both  $L_\infty$  and  $L_1$  constraints. Therefore, it's likely that a more intricate set of energy matrices is needed to proceed in this direction.

When viewed from the perspective of combinatorial preconditioning, we were able to solve multicommodity electrical flows by adding  $\epsilon$  slack to the edges, thus 'fixing' the condition number. For the purpose of obtaining  $1 \pm \epsilon$  approximations to combinatorial problems, this does not modify the solution by too much. However, it is unlikely to be applicable inside algorithms whose dependency on  $\epsilon$  is  $O(\log(1/\epsilon))$ , as very little slack can be added onto the edges. As a result, we believe that obtaining fast solvers for the class of matrices that arise from quadratically coupled flows is an interesting direction for future work.

## References

- [AK07] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In *STOC*, pages 227–236, 2007. 6, 6, 6
- [AT11] Haim Avron and Sivan Toledo. Effective stiffness: Generalizing effective resistance sampling to finite element matrices. *CoRR*, abs/cs/1110.4437, 2011. 1.2
- [Axe94] Owe Axelsson. *Iterative Solution Methods*. Cambridge University Press, New York, NY, 1994. 4.4
- [BHV04] Erik G. Boman, Bruce Hendrickson, and Stephen A. Vavasis. Solving elliptic finite element systems in near-linear time with support preconditioners. *CoRR*, cs.NA/0407022, 2004. 1.2, 3, 4.2
- [CKM<sup>+</sup>11] Paul Christiano, Jonathan A. Kelner, Aleksander Mądry, Daniel Spielman, and Shang-Hua Teng. Electrical Flows, Laplacian Systems, and Faster Approximation of Maximum Flow in Undirected Graphs. In *Proceedings of the 43<sup>rd</sup> ACM Symposium on Theory of Computing (STOC)*, 2011. 1.2, 1.2, 2.1, 3, 2, 6, A
- [CSRL01] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001. 7.1
- [DS07] Samuel I. Daitch and Daniel A. Spielman. Support-graph preconditioners for 2-dimensional trusses. *CoRR*, abs/cs/0703119, 2007. 1.2
- [DS08] Samuel I. Daitch and Daniel A. Spielman. Faster approximate lossy generalized flow via interior point algorithms. *CoRR*, abs/0803.0988, 2008. 1.1, 3
- [Fle00] Lisa K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM Journal on Discrete Mathematics*, 13:505–520, 2000. 1.1
- [GK98] Naveen Garg and Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *In Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 300–309, 1998. 1.1

- 
- [KMP10] Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching optimality for solving SDD systems. *CoRR*, abs/1003.2958, 2010. 4.5
  - [KMP11] Ioannis Koutis, Gary L. Miller, and Richard Peng. Solving sdd linear systems in time  $\tilde{O}(m \log n \log(1/\epsilon))$ . *CoRR*, abs/1102.4842, 2011. 4.5
  - [KS80] Ross Kindermann and J. Laurie Snell. *Markov random fields and their applications*. American Mathematical Society, Providence, R.I., 1980, 1980. 8
  - [LMP<sup>+</sup>91] Tom Leighton, Fillia Makedon, Serge Plotkin, Clifford Stein, Eva Tardos, and Spyros Tragoudas. Fast approximation algorithms for multicommodity flow problems. In *JOURNAL OF COMPUTER AND SYSTEM SCIENCES*, pages 487–496, 1991. 1.1, 1, 8
  - [Mad10] Aleksander Madry. Faster approximation schemes for fractional multicommodity flow problems via dynamic graph algorithms. In *STOC '10: Proceedings of the 42nd ACM symposium on Theory of computing*, pages 121–130, New York, NY, USA, 2010. ACM. 1.1
  - [RW66] B. Rothschild and A. Whinston. Feasibility of two commodity network flows. *Operations Research*, 14(6):pp. 1121–1129, 1966. 1.1, 1.2
  - [Saa96] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. <http://www-users.cs.umn.edu/~saad/books.html>, 1996. 4.4
  - [ST06] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *CoRR*, abs/cs/0607105, 2006. 1.1, 4.5
  - [SZS<sup>+</sup>08] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1068–1080, 2008. 8
  - [Vai89] P. M. Vaidya. Speeding-up linear programming using fast matrix multiplication. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 332–337, Washington, DC, USA, 1989. IEEE Computer Society. 1.1
  - [Vai91] Preadeep M. Vaidya. Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners. A talk based on this manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computation, October 1991. 1.1
  - [YL93] Shing-Tung Yau and Ya Yan Lu. Reducing the symmetric matrix eigenvalue problem to matrix multiplications. *SIAM J. Sci. Comput.*, 14:121–136, January 1993. 6

## A Obtaining Almost Optimal Electrical Flow from Potentials

This approximate solve then allows us to compute electrical flows in a way analogous to theorem 2.3 of [CKM<sup>+</sup>11]. Pseudocode of the algorithm is shown in Algorithm 6.

The algorithm makes calls to MAKEKIRCHHOFF, which converts a flow  $\hat{\mathbf{f}}$  that doesn't satisfy conservation of flow at vertices in  $W$  to one that does without too much increase in congestion. Its properties are proven in Lemma A.1.

We start by bounding the errors incurred by MAKEKIRCHHOFF

**Lemma A.1** *Given a capacitated graph  $G = (V, E, \mathbf{u})$ , a set of demands  $\mathbf{d}$  a flow  $\hat{\mathbf{f}}$  such that  $\|\Gamma^T \hat{\mathbf{f}}\|_\infty \leq \gamma$ . MAKEKIRCHHOFF( $G, \mathbf{d}, \hat{\mathbf{f}}$ ) returns a flow  $\tilde{\mathbf{f}}$  such that:*

1.  $\Gamma^T \tilde{\mathbf{f}} = \mathbf{d}$ .
2. For all edges  $e \in E$ ,  $\|\hat{\mathbf{f}}_e - \tilde{\mathbf{f}}_e\|_\infty \leq m\gamma$ .

**Proof** Part 1 follows from the construction for each vertex  $u \neq r$  and  $\mathbf{1}^T \mathbf{d} = 0$ . To bound the extra congestion, note that each edge in the tree is used in at most  $n < m$  paths that route at most  $\gamma$  units of flow each. This gives an extra congestion of at most  $m\gamma$  and therefore Part 2. ■

**Proof of Theorem 4.7:**

We start off by showing that  $\tilde{\lambda}$  does not differ by much from  $\bar{\lambda} = \sqrt{\phi^T \mathcal{L} \phi}$ :

---

**Algorithm 6** Algorithm for computing near-optimal electrical potential/flow pair

---

QUADRATICALLYCOUPLEDFLOW

Input: Graph  $G = (V, E)$  with corresponding edge-vertex incidence matrix  $\mathbf{B}$ . Demand vector  $\mathbf{d}$  Energy matrices for the edges  $\mathbf{P}$  along with bounds  $U, \kappa$  such that  $I \preceq \mathbf{P} \preceq U \cdot I$  and  $\lambda_{\max}(\mathbf{P}(e)) \preceq \kappa \lambda_{\min}(\mathbf{P}(e))$ . Error bound  $\delta$ .

Output: Approximate vertex potentials  $\tilde{\phi}$  and flow  $\tilde{\mathbf{f}}$  that satisfies the constraints of Theorem

- 1: Let  $\mathcal{L}$  be an implicit representation of the matrix  $\Gamma \mathbf{P} \Gamma$
  - 2: Use PRECONCHEBY to find  $\tilde{\mathbf{x}} = \mathcal{L}^+ \mathbf{d}$  with error of  $\epsilon = \frac{\delta}{5m^6 k^2 U^4}$
  - 3:  $\tilde{\lambda} \leftarrow \sqrt{\tilde{\mathbf{x}}^T \mathcal{L} \tilde{\mathbf{x}}}$
  - 4:  $\tilde{\phi} = \frac{1}{\tilde{\lambda}} \tilde{\mathbf{x}}$
  - 5: Compute ohmic flow  $\hat{\mathbf{f}} = \mathbf{P}^{-1} \Gamma \tilde{\mathbf{x}}$
  - 6: **for**  $i = 1 \dots k$  **do**
  - 7:    $\tilde{\mathbf{f}}_i \leftarrow \text{MAKEKIRCHHOFF}(G, \mathbf{d}_i, \hat{\mathbf{f}}_i)$
  - 8: **end for**
  - 9: **return**  $\tilde{\phi}, \tilde{\mathbf{f}}$
- 

---

**Algorithm 7** Algorithm for making an arbitrary flow Kirchhoff

---

MAKEKIRCHHOFF

Input: Graph  $G = (W \subseteq V, E)$  with demands  $\mathbf{d}$ . A Ohmic flow  $\mathbf{f}$ .

Output: A flow that satisfies conservation of flow at all vertices in  $W$

- 1:  $\tilde{\mathbf{f}} \leftarrow \mathbf{f}$
  - 2: Pick any vertex  $r$
  - 3: compute a spanning tree,  $T$
  - 4: **for** each vertex  $u \neq r$  **do**
  - 5:   route  $\mathbf{d}_u - (\Gamma \tilde{\mathbf{f}})_u$  units of flow from  $u$  to  $r$  along  $T$
  - 6: **end for**
  - 7: **return**  $\tilde{\mathbf{f}}$
-

---

**Lemma A.2**

$$|\tilde{\lambda} - \bar{\lambda}| \leq \epsilon \bar{\lambda}$$

*Proof*

$$\begin{aligned} |\tilde{\lambda} - \bar{\lambda}| &= ||\tilde{\mathbf{x}}|_{\mathcal{L}} - |\bar{\mathbf{x}}|_{\mathcal{L}}| \\ &\leq ||\tilde{\mathbf{x}} - \bar{\mathbf{x}}|_{\mathcal{L}} \quad \text{Since } ||\cdot|_{\mathcal{L}} \text{ is a norm} \\ &\leq \epsilon ||\bar{\mathbf{x}}|_{\mathcal{L}} = \epsilon \bar{\lambda} \end{aligned} \tag{1.85}$$

■(Lemma A.2)

Algebraic manipulations of this give:

**Corollary A.3**

$$(1 - \epsilon) \bar{\lambda} \leq \tilde{\lambda} \leq (1 + \epsilon) \bar{\lambda} \tag{1.86}$$

$$(1 - 2\epsilon) \frac{1}{\bar{\lambda}} \leq \frac{1}{\tilde{\lambda}} \leq (1 + 2\epsilon) \frac{1}{\bar{\lambda}} \tag{1.87}$$

This allows us to show that  $\tilde{\mathbf{f}} = \tilde{\lambda} \mathbf{P}^{-1} \Gamma \tilde{\mathbf{x}}$  does not differ from  $\bar{\mathbf{f}} = \bar{\lambda} \mathbf{P}^{-1} \Gamma \bar{\mathbf{x}}$  by too much:

**Lemma A.4**

$$\begin{aligned} \|\hat{\mathbf{f}} - \bar{\mathbf{f}}\|_{\mathbf{P}} &\leq \sqrt{(\tilde{\mathbf{f}} - \bar{\mathbf{f}})^T \mathbf{P} (\tilde{\mathbf{f}} - \bar{\mathbf{f}})} \\ &\leq 4\epsilon \bar{\lambda}^2 \\ &= 4\epsilon \mathcal{E}_{\mathcal{F}}(\mathbf{P}) \end{aligned} \tag{1.88}$$

*Proof*

$$\begin{aligned} \|\hat{\mathbf{f}} - \bar{\mathbf{f}}\|_{\mathbf{P}} &= \|\mathbf{P}^{-1} \Gamma \tilde{\mathbf{x}} - \mathbf{P}^{-1} \Gamma \bar{\mathbf{x}}\|_{\mathbf{P}} \\ &= \|\tilde{\mathbf{x}} - \bar{\mathbf{x}}\|_{\mathcal{L}} \quad \text{By definition of } \mathcal{L} \\ &\leq \epsilon ||\bar{\mathbf{x}}|_{\mathcal{L}} \end{aligned} \tag{1.89}$$

■(Lemma A.4)

This in turn gives bounds on the maximum entry-difference of  $\hat{\mathbf{f}}$  and  $\bar{\mathbf{f}}$ :

**Corollary A.5**

$$||\tilde{\mathbf{f}} - \bar{\mathbf{f}}|_{\infty} \leq \epsilon \bar{\lambda}$$

*Proof* Follows from  $\mathbf{P} \succeq \mathbf{I}$  and  $||\mathbf{v}|_{\infty} \leq ||\mathbf{v}|_2$  for any vector  $\mathbf{v}$ . ■(Corollary A.5)

Since  $\bar{\mathbf{f}}$  meets the demands at each vertex, the amount that  $\hat{\mathbf{f}}$  does not meet the demand by is at most  $\epsilon m k \bar{\lambda}$ . Lemma A.1 gives that the difference along each edge after running MAKEKIRCHHOFF is at most  $\epsilon m^2 k^2 \bar{\lambda}$ . The following two bounds are immediate consequences of this:

1.

$$\begin{aligned} \|\tilde{\mathbf{f}} - \bar{\mathbf{f}}\|_{\mathbf{P}} &\leq \|\tilde{\mathbf{f}} - \hat{\mathbf{f}}\|_{\mathbf{P}} + \|\hat{\mathbf{f}} - \bar{\mathbf{f}}\|_{\mathbf{P}} \\ &\leq \epsilon U m^2 k^2 \bar{\lambda} + \epsilon \bar{\lambda} \quad \text{Since } \mathbf{P} \preceq U \mathbf{I} \\ &\leq 2\epsilon U^2 m^2 k^2 \bar{\lambda} \end{aligned} \tag{1.90}$$

2.

$$\begin{aligned} \left| \sqrt{\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P})} - \sqrt{\mathcal{E}_{\bar{\mathbf{f}}}(\mathbf{P})} \right| &= \left| \|\tilde{\mathbf{f}}\|_{\mathbf{P}} - \|\bar{\mathbf{f}}\|_{\mathbf{P}} \right| \\ &\leq \|\tilde{\mathbf{f}} - \bar{\mathbf{f}}\|_{\mathbf{P}} \leq 2\epsilon U^2 m^2 k^2 \bar{\lambda} \end{aligned} \quad (1.91)$$

Which in turn implies that  $\sqrt{\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P})} \leq (1 + 5\epsilon U^2 m^2 k^2) \sqrt{\mathcal{E}_{\bar{\mathbf{f}}}(\mathbf{P})}$  when  $\epsilon U^2 m^2 k^2 < 0.01$ .

This enables us to derive the overall bound for the  $L_2$  energy difference of the two flows. We first need one more identity about the pointwise difference  $|\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P}, e) - \mathcal{E}_{\bar{\mathbf{f}}}(\mathbf{P}, e)|$ . Since  $\mathbf{P}(e)$  is positive definite, we may write it as  $\mathbf{Q}(e)^T \mathbf{Q}(e)$ . Then the difference in energy can be written as:

$$\begin{aligned} &|\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P}, e) - \mathcal{E}_{\bar{\mathbf{f}}}(\mathbf{P}, e)| \\ &= \left| \|\mathbf{Q}(e)\tilde{\mathbf{f}}(e)\|_2^2 - \|\mathbf{Q}(e)\bar{\mathbf{f}}(e)\|_2^2 \right| \\ &= \left| (\mathbf{Q}(e)\tilde{\mathbf{f}}(e) + \mathbf{Q}(e)\bar{\mathbf{f}}(e))^T (\mathbf{Q}(e)\tilde{\mathbf{f}}(e) - \mathbf{Q}(e)\bar{\mathbf{f}}(e)) \right| \quad \text{By applying } a^2 - b^2 = (a+b)(a-b) \text{ entry-wise} \\ &\leq \|\mathbf{Q}(e)\tilde{\mathbf{f}}(e) + \mathbf{Q}(e)\bar{\mathbf{f}}(e)\|_2 \|\mathbf{Q}(e)\tilde{\mathbf{f}}(e) - \mathbf{Q}(e)\bar{\mathbf{f}}(e)\|_2 \quad \text{by the Cauchy-Schwarz inequality} \\ &= \|\tilde{\mathbf{f}}(e) + \bar{\mathbf{f}}(e)\|_{\mathbf{P}(e)} \|\tilde{\mathbf{f}}(e) - \bar{\mathbf{f}}(e)\|_{\mathbf{P}(e)} \end{aligned} \quad (1.92)$$

Since  $\mathbf{P}$  is a block diagonal matrix divided by the edges, we have  $\|\mathbf{v}\|_{\mathbf{P}} = \sum_e \|\mathbf{v}(e)\|_{\mathbf{P}(e)}$  for any vector  $\mathbf{v}$ . Therefore:

$$\begin{aligned} \|\tilde{\mathbf{f}}(e) + \bar{\mathbf{f}}(e)\|_{\mathbf{P}(e)} &\leq \|\tilde{\mathbf{f}} + \bar{\mathbf{f}}\|_{\mathbf{P}} \\ &\leq \sqrt{4\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P})} \quad \text{since } (a+b)^2 \leq 2(a^2 + b^2) \text{ and } \mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P}) \leq \mathcal{E}_{\bar{\mathbf{f}}}(\mathbf{P}) \end{aligned} \quad (1.93)$$

$$\begin{aligned} \|\tilde{\mathbf{f}}(e) - \bar{\mathbf{f}}(e)\|_{\mathbf{P}(e)} &\leq \|\tilde{\mathbf{f}} - \bar{\mathbf{f}}\|_{\mathbf{P}} \\ &\leq (1 + 5\epsilon U^2 m^2 k^2) \bar{\lambda} \end{aligned} \quad (1.94)$$

Combining them gives:

$$\begin{aligned} |\mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P}, e) - \mathcal{E}_{\bar{\mathbf{f}}}(\mathbf{P}, e)| &\leq (5\epsilon U^2 m^2 k^2 \bar{\lambda}) ((1 + 5\epsilon U^2 m^2 k^2) \bar{\lambda}) \\ &\leq 10\epsilon U^2 m^2 k^2 \mathcal{E}_{\tilde{\mathbf{f}}}(\mathbf{P}) \quad \text{When } 5\epsilon U^2 m^2 k^2 < 1 \end{aligned} \quad (1.95)$$

It can be checked that when  $\delta < 0.1$ , setting  $\epsilon = \frac{\delta}{2m^2 k^2 U}$  satisfies the condition for the last inequality, as well as bounding the last term by  $\delta$ . For this setting, we have  $\log(1/\epsilon) \leq O(\log(Umk/\delta))$ .

■(Theorem 4.7)

## B Maximum Weighted Multicommodity Flow

The maximum weighted multicommodity flow problem is a related problem that maximizes  $\sum_i \lambda_i \mathcal{F}_i$  for a series of weights  $\lambda \in \mathbb{R}_+^k$ , where  $\mathcal{F}_i$  is the amount of flow of commodity  $i$  that's routed between the demand pairs. In this section we give an overview of how to extend our algorithm to this problem. However, in order to simplify presentation we assume that the solves are exact. Errors from these solves can be analyzed in steps similar to those in our algorithm for maximum concurrent flow.

In our notation, the problem can be formulated as finding flows  $\mathbf{f}_1 \dots \mathbf{f}_k$  such that  $\mathbf{B}\mathbf{f}_i = \mathcal{F}_i \mathbf{d}_i$  and maximizing  $\sum_i \lambda_i \mathcal{F}_i$ . Furthermore it can be reduced to the decision problem of finding  $\mathbf{f}_i, \mathcal{F}_i$  such that



$\sum_i \mathcal{F}_i = 1$ . Note that due to the undirected nature of the flow, we can allow  $\mathcal{F}_i$  to be negative since  $\lambda_i \geq 0$  means negating the flow of the  $i^{th}$  commodity can only improve the overall objective. Once we introduce the  $k$  additional variables  $\mathcal{F}_1 \dots \mathcal{F}_k$ , the maximum weighted multicommodity flow problem can be formulated as:

$$\begin{aligned} & \text{maximize: } \sum_i \mathcal{F}_i \\ & \text{subject to: } \sum_{i=1}^k |\mathbf{f}_i(e)| \leq u(e) & \forall e \in E \\ & \mathbf{B}^T \mathbf{f}_i = \mathcal{F}_i \mathbf{d}_i & \forall 1 \leq i \leq k \end{aligned}$$

By binary search and appropriate scaling of  $\mathbf{d}_i$ , it suffices to check whether there is a solution  $(\mathbf{f}, \mathcal{F})$  where  $\mathbf{1}^T \mathcal{F} = 1$ . Furthermore we can use  $\mathbf{D}$  to denote the  $kn \times k$  matrix with column  $i$  being  $\mathbf{d}_i$  extended to all  $kn$  vertex/commodity pairs. Then if we let  $\mathcal{F}$  denote the vector containing all the flow values, the corresponding quadratically coupled flow problem becomes:

$$\begin{aligned} & \text{minimize: } \mathcal{E}(\mathbf{P}, \mathbf{f}) \\ & \text{subject to: } \Gamma^T \mathbf{f} = \mathbf{D} \mathcal{F} \\ & \mathbf{1}^T \mathcal{F} = 1 \end{aligned}$$

As before, we let  $\mathcal{L} = \Gamma^T \mathbf{P} \Gamma$  and define the following quantities:

$$\lambda = \frac{1}{\mathbf{1}^T (\mathbf{D} \mathcal{L} + \mathbf{D}) + \mathbf{1}} \quad (2.96)$$

$$\bar{\mathcal{F}} = \lambda (\mathbf{D} \mathcal{L} + \mathbf{D}) + \mathbf{1} \quad (2.97)$$

$$\bar{\phi} = \mathcal{L} + \mathbf{D} \bar{\mathcal{F}} \quad (2.98)$$

$$\bar{\mathbf{f}} = \mathbf{P}^{-1} \Gamma \bar{\phi} \quad (2.99)$$

Note that the matrix  $\mathbf{D} \mathcal{L} + \mathbf{D}$  can be explicitly computed using  $k^2$  solves, and the resulting  $k$ -by- $k$  matrix can be inverted using direct methods in  $O(k^\omega)$  time. The following lemmas similar to the ones about quadratically coupled flows shown in Section 4 can be checked in an analogous way.

### Lemma B.1

$$\Gamma^T \bar{\mathbf{f}} = \mathbf{D} \bar{\mathcal{F}}$$

*Proof*

$$\begin{aligned} \Gamma^T \bar{\mathbf{f}} &= \Gamma^T \mathbf{P}^{-1} \Gamma \bar{\phi} \\ &= \mathcal{L} \bar{\phi} \\ &= \mathcal{L} \mathcal{L} + \mathbf{D} \bar{\mathcal{F}} \\ &= \mathbf{D} \bar{\mathcal{F}} \end{aligned} \quad (2.100)$$

■

---

**Lemma B.2**

$$\mathcal{E}(\mathbf{P}, \bar{\mathbf{f}}) = \mathcal{E}(\bar{\phi}) = \lambda$$

*Proof*

$$\begin{aligned}
\mathcal{E}(\mathbf{P}, \bar{\mathbf{f}}) &= \bar{\mathbf{f}}^T \mathbf{P} \bar{\mathbf{f}} \\
&= \bar{\phi}^T \Gamma^T \mathbf{P}^{-1} \mathbf{P} \mathbf{P}^{-1} \Gamma \bar{\phi} \\
&= \bar{\phi}^T \mathcal{L} \bar{\phi} \\
&= \bar{\mathcal{F}} \mathbf{D}^T \mathcal{L}^+ \mathcal{L} \mathbf{D} \bar{\mathcal{F}} \\
&= \bar{\mathcal{F}} \mathbf{D}^T \mathcal{L} \mathbf{D} \bar{\mathcal{F}} \\
&= \lambda^2 \mathbf{1}^T (\mathbf{D} \mathcal{L}^+ \mathbf{D})^+ (\mathbf{D}^T \mathcal{L} \mathbf{D}) (\mathbf{D} \mathcal{L}^+ \mathbf{D})^+ \mathbf{1} \\
&= \lambda^2 \frac{1}{\lambda} = \lambda
\end{aligned} \tag{2.101}$$

■

**Lemma B.3** For any  $(\mathbf{f}, \mathcal{F})$  pair that satisfies  $\Gamma^T \mathbf{f} = \mathbf{D} \mathcal{F}$  and  $\mathbf{1}^T \mathcal{F} = 1$ , we have:

$$\mathcal{E}(\mathbf{P}, \mathbf{f}) \geq \lambda$$

*Proof* Since  $\mathcal{E}(\bar{\phi}) = \lambda$  by Lemma B.2, it suffices to show  $\mathcal{E}(\mathbf{P}, \mathbf{f}) \mathcal{E}(\bar{\phi}) \geq \lambda^2$ .

$$\begin{aligned}
\mathcal{E}(\mathbf{P}, \mathbf{f}) \mathcal{E}(\bar{\phi}) &= (\bar{\mathbf{f}}^T \mathbf{P} \bar{\mathbf{f}}) ((\Gamma \bar{\phi})^T \mathbf{P}^{-1} (\Gamma \bar{\phi})) && \text{By definition of } \bar{\mathbf{f}} \text{ in Equation 2.99} \\
&\geq (\bar{\mathbf{f}}^T \Gamma \bar{\phi})^2 && \text{By Cauchy-Schwarz inequality} \\
&= (\mathcal{F}^T \mathbf{D}^T \bar{\phi})^2 && \text{Since } \Gamma^T \mathbf{f} = \mathbf{D} \mathcal{F} \\
&= (\mathcal{F}^T \mathbf{D}^T \mathcal{L}^+ \mathbf{D} \bar{\mathcal{F}})^2 && \text{By definition of } \bar{\phi} \text{ in Equation 2.98} \\
&= (\mathcal{F}^T (\lambda \mathbf{1}))^2 && \text{By definition of } \bar{\mathcal{F}} \text{ in Equation 2.97} \\
&= \lambda^2 && \text{Since } \mathbf{1}^T \mathcal{F} = 1
\end{aligned} \tag{2.102}$$

■

Furthermore, Lemma 5.2, Part 3, which is crucial for showing an increase in the minimum energy of the quadratically coupled flow, still holds. Specifically, Equations 5.40 and 5.41 adapts readily with the  $\phi$ . Therefore generalizing Theorem 5.1 and combining it with Theorem 6.4 gives an analogous result for approximating the maximum weighted multicommodity flow problem:

**Theorem B.4** Given an instance of the maximum weighted multicommodity flow problem with  $k$  commodities on a graph with  $m$  edges and an error parameter  $\epsilon > 0$ . A solution with weight at least  $(1 - \epsilon)$  of the maximum can be produced in  $\tilde{O}(m^{4/3} \mathbf{poly}(k, \epsilon^{-1}))$  time.